

MRDOC Software Configuration Management Plan

Exhibits I, II and III

Preliminary
October 20, 2000

AUTHORIZED by CM when under FORMAL Configuration Control	
Date	Signature
10/31/00	/s/ Robert H. Van Niel



Prepared For:
National Aeronautics and Space Administration
John H. Glenn Research Center
Microgravity Science Division
Cleveland, Ohio 44135



Prepared By:
Federal Data Corporation
Under Contract NAS3-99155
2001 Aerospace Parkway
Brook Park, Ohio 44142

PREFACE

The National Aeronautics and Space Administration (NASA) is developing a modular, multi-user experimentation facility for conducting fluid physics and combustion science experiments in the microgravity environment of the International Space Station (ISS). This facility, called the Fluids and Combustion Facility (FCF), consists of three test platforms: the Fluids Integrated Rack (FIR), the Combustion Integrated Rack (CIR), and the Shared Accommodations Rack (SAR).

This MRDOC Software Configuration Management Plan has been prepared by Federal Data Corporation's Major Products Group in accordance with the requirements specified in the Statement of Work (SOW), Contract No. NAS3-99155, entitled Microgravity Research, Development and Operations Contract (MRDOC). This plan is consistent with, and implements the guidelines of, MIL-STD-973, MIL-STD-498, DOD STD-2167A, Industry Standard IEEE 12207, Configuration Management, NAS-STD-2100-91, NASA Software Documentation Standard, and ISO 10007, Quality Management – Guidelines for Configuration Management.

MRDOC SOFTWARE CONFIGURATION MANAGEMENT PLAN EXHIBITS I, II and III

Prepared By: /s/ Alan Germovsek **Date:** 10/30/00
Alan Germovsek
Configuration Management
Federal Data Corporation

Reviewed By: /s/ Jon Wetherholt **Date:** 10/30/00
Jon Wetherholt
Product Development Team Lead
Combustion Integrated Rack Development Program
Analex Corporation

Concurred By: /s/ Andrew M. Peddie **Date:** 10/30/00
Andrew M. Peddie
FCF Deputy Director
Federal Data Corporation

Concurred By: /s/ Christopher J. Pestak **Date:** 10/30/00
Christopher J. Pestak
FCF Director
Analex Corporation

Approved By: /s/ James A. Drury **Date:** 10/30/00
James A. Drury
Configuration Manager
Federal Data Corporation

Approved By: /s/ Tom Ziemianski **Date:** 10/30/00
Tom Ziemianski
Product Assurance
Hernandez Engineering, Inc,

Approved By: /s/ Brian Finley **Date:** 10/30/00
Brian Finley
PI Experiments Director
Federal Data Corporation

Approved By: /s/ Michael Korba

Michael Korba
MRDOC Deputy Program Manager
Federal Data Corporation

Date: 10/30/00

Approved By: /s/ Dr. Guion Bluford

Dr. Guion Bluford
MRDOC Program Manager
Federal Data Corporation

Date: 10/30/00

Approved By: _____

Robert Zurawski
Project Manager, Fluids and Combustion Facility (FCF)
NASA Glenn

Date: _____

REVISION PAGE
MRDOC SOFTWARE CONFIGURATION MANAGEMENT PLAN
EXHIBITS I, II and III

Revision	Date	Description of Change or ECO's/ECP's Incorporated	Verification and Date
Preliminary	10/20/00	Initial	10/30/00

TABLE OF CONTENTS

1.0	Introduction	1
1.1	Objective.....	1
1.2	Purpose.....	1
	1.2.1 Effective Date.....	2
1.3	Scope.....	2
1.4	Approach.....	2
1.5	System Overview	3
1.6	Order of Precedence for Verification Requirements.....	5
1.7	Acronyms	5
1.8	Definitions.....	5
2.0	Documents	6
2.1	Order of Precedence for Documents.....	6
2.2	Applicable Documents.....	6
2.3	Reference Documents.....	7
2.4	Source of Documents.....	8
2.5	Tailoring.....	8
3.0	Software Configuration Management	9
3.1	SCM Organization.....	9
3.2	SCM Responsibilities	9
	3.2.1 Configuration Identification.....	9
	3.2.2 Configuration Control.....	11
	3.2.3 Configuration Status Accounting (CSA).....	11
	3.2.4 Configuration Audits.....	11
	3.2.5 Training	12
	3.2.6 Program Configuration Manager	12
	3.2.7 SCM Project Specialist	12
	3.2.8 Library Project Control Specialist	12
	3.2.9 Software Administrator.....	13
3.3	Project organizational structure.....	13
	3.3.1 Control/Review Board	15
3.4	Relationship of CM to the Software Process Life Cycle	17
4.0	CM phasing and Milestones	19
4.1	REQUIREMENTS DEFINITION	19
4.2	DESIGN.....	21
4.3	DEVELOPMENT.....	21
	4.3.1 Requirements Definition	21

4.3.2	Project Planning and Oversight	22
4.3.3	Establishment of Software Development Environment	22
4.3.4	System Requirements Analysis	23
4.3.5	System Design	23
4.3.6	Software Requirements Analysis	24
4.3.7	Software Design	25
4.3.8	Software Implementation and Unit Testing	25
4.3.9	Unit Integration and Testing	26
4.3.10	CSCI Qualification Testing	27
4.3.11	CSCI/Hardware Configuration Item (HWCI) Integration and Testing	28
4.3.12	System Qualification Testing	28
4.3.13	Software Use Preparation	29
4.3.14	Software Transition Preparation	30
4.4	INSTALLATION AND DEPLOYMENT	31
4.5	OPERATIONS AND SUPPORT	31
5.0	DATA MANAGEMENT	32
5.1	DATA DISTRIBUTION AND ACCESS	32
5.2	AUTOMATED PROCESSING AND SUBMITTAL OF DATA	32
5.3	INTERACTIVE ACCESS TO DIGITAL DATA	33
5.4	STATUS REPORTING	34
5.5	DATA SECURITY AND CLASSIFICATION MANAGEMENT	34
6.0	Configuration Identification	35
6.1	Documentation Identification	36
6.1.1	Document Types	38
6.2	Document Release	40
6.3	Software Document Revisions	40
6.3.1	Software identification	41
6.4	Software Identification	41
6.4.1	Software Modules	41
6.4.1.1	Preliminary Versions	41
6.4.1.2	Alpha Versions	41
6.4.1.3	Beta Versions	42
6.4.1.4	Baseline Versions	42

6.4.2	Software Subsystems	42
6.4.2.1	Beta Versions	42
6.4.2.2	Release Versions	43
6.4.3	Licensed Software.....	43
6.4.4	Purchased Commercial Software.....	43
6.5	Change Control Form Identification	43
6.6	Configuration Baselines.....	44
6.6.1	Functional Baseline	44
6.6.2	Allocated Baseline	44
6.6.3	Developmental Baseline.....	44
6.6.4	Product Baseline	46
6.6.5	Production Baseline	46
6.6.6	Configuration Releases	46
6.6.7	Physical Item Hierarchy	46
6.7	Software Libraries	47
6.7.1	Configuration Control of Libraries	47
6.7.2	Storage, Handling, and Release of Project Media	48
6.7.3	Software Development Library	49
6.7.4	Software Integration Library	49
6.7.5	Production Software Library	49
6.7.6	Software Repository.....	50
7.0	Interface management	51
7.1	Organizational Interface	51
7.2	Phase Interface	52
7.3	Software Interfaces	52
7.4	Hardware Interfaces.....	52
8.0	Configuration Control Processes	53
8.1	Software Configuration Management Specialist.....	53
8.2	Closed-Looped Configuration Control Process	54
8.3	Problem Report	54
8.3.1	Customer Initiated Problem Reporting and Corrective Action	55
8.4	Urgent ECO/ECP/RDW Processing.....	55
8.5	Fast Track ECO/ECP/RDW Processing	57
8.6	Change Request Initiation.....	58
8.7	Change Control Organization, Authorities and Activities.....	59

8.8	Change Control Boards.....	59
9.0	Configuration Status Accounting.....	62
9.1	Policy and Procedure.....	62
9.2	Status Accounting Control Items	62
	9.2.1 Document Status Listing (DSL)	63
	9.2.2 Problem Reporting and Tracking.....	63
10.0	Configuration Management Design Reviews and Audits	65
10.1	Reviews.....	65
	10.1.1 Software Specification Review (SSR)	65
	10.1.2 Preliminary Design Review (PDR).....	65
	10.1.3 Critical Design Review (CDR).....	65
	10.1.4 Verification and Test Review (VTR).....	66
10.2	Configuration Audits	66
	10.2.1 Functional Configuration Audit (FCA).....	66
	10.2.2 Physical Configuration Audit (PCA)	67
10.3	Continuing Performance Audits and Surveillance	67
	10.3.1 Audits and Reviews of CM	67
	10.3.1.1 SCM Audits.....	67
	10.3.1.2 SCM Reviews.....	68
11.0	Subcontractor/Vendor Control.....	69
11.1	Subcontractor Reviews and Audits	69
11.2	Major Subcontractors	69

LIST OF APPENDICES

APPENDIX A	acronyms and abbreviations	71
A.1	Scope.....	71
A.2	List of acronyms and abbreviations.....	71
APPENDIX B	Definitions	74
B.1	Scope.....	74
B.2	List of definitions.	74

LIST OF FIGURES

Figure 1	Overview of SCM Functions	4
Figure 2	SCM Interface Within MRDOC Organization	10
Figure 3	CM Organizational Structure.....	10
Figure 4	MRDOC Project/SCM Organization Structure	14
Figure 5	MRDOC Software Configuration Control Board	16
Figure 6	Configuration Data Management	18
Figure 7	SCM Milestones.....	20
Figure 8	MRDOC Software Documentation Preparation and Review Flow	40
Figure 9	System/Software Development Life Cycle	45
Figure 10	Hierarchical Structure for Software	47
Figure 11	Urgent ECO/ECP/RDW Processing	56
Figure 12	Change Control Process Class II ECO/ECP Minor RDW	57
Figure 13	Change Control Process Class I ECO/ECP Major RDW.....	58

1.0 INTRODUCTION

This Software Configuration Management (SCM) Plan (SCMP) describes the SCM organization and practices applied consistently and uniformly throughout the life cycle for Computer Software Configuration Items (CSCIs) that are developed or maintained by Federal Data Corporation's (FDC) Major Program Sector.

SCM is the process used during software development and maintenance to identify, control, and report functional and physical configurations of software products (e.g., source code, executable code, databases, test scenarios and data, and documentation).

1.1 Objective.

The objective of SCM is to establish and manage the developmental configuration and control procedures for the Computer Software Configuration Items (CSCI), thus ensuring the integrity of the product. SCM consists of identifying the system configuration at discrete points in time for methodically controlling and maintaining the integrity and traceability of the software throughout the project. SCM is practiced at FDC over the entire lifetime of the software process in order to reduce errors and version conflict and maximize productivity during the course of development and maintenance for the MRDOC Program. As CSCIs are produced, they are reviewed and added to our project database. The database holds all of our current and past code and documentation for the MRDOC Program. CSCIs may be programs, program code, documentation, and data. FDC's SCM process follows five standard tasks:

- * Identification of items in the software configuration
- * Version control
- * Change control
- * Configuration auditing
- * Configuration status accounting (status reporting)

This plan is a functional part of the Software Development Plan (SDP). A separate Software Configuration Management Plan (SCMP) has been produced on the grounds of establishing a common method of identifying, controlling and recording the software baselines and developmental configuration for each CSCI.

1.2 Purpose.

This plan identifies, defines and establishes the overall methods and procedures used to perform software configuration management during the development and maintenance of all software products for the MRDOC Program. This document provides the basis for all software configuration management activities under this program. The plan describes the organizational responsibilities, SCM requirements, methods, basic procedures and vendor requirements necessary to provide configuration control of the documents and the software, as well as the accurate reporting of configuration status. This document is prepared in accordance with Contract NAS3-99155, in accordance with the guidelines established in NAS-STD-2100-91,

NASA Software Documentation Standard, and ISO 10007, Quality Management – Guidelines for Configuration Management.

1.2.1 Effective Date.

This SCM Plan is effective as of the date of publication and will remain in effect until canceled or modified by MRDOC Program management or a designated representative. Personnel assigned to or in support of the MRDOC Program will adhere to the provisions of this plan and its underlying purposes, to ensure the production of the contracted product, the incorporation of agreed to changes and the identification and maintenance of valid documentation.

1.3 Scope

The SCM discipline is applied to those Configuration Items (CIs) and Computer Software Configuration Items (CSCIs), for which FDC's Major Program Sector has development and/or maintenance responsibilities for on the MRDOC Exhibits. The SCM organization implements the processes described within this plan to ensure that products developed are correct, consistent, complete, and compliant with governing policies.

This Plan applies to all Federal Data Corporation elements and FDC subcontractors engaged in development, build, maintenance support, integration and operations where CSCIs are applicable. This document establishes and defines the responsibilities, software documentation requirements, and the related processes for controlling science requirements, CI/CSCI technical requirements, internal and external interfaces, and subcontractor/vendor requirements. SCM practices will be imposed on systems, subsystems, related ground support equipment, and all documentation, to ensure accurate reporting of software configuration status throughout all phases of the MRDOC Program.

The plan describes the organizational relationship of SCM to total program management, defines the methods and documentation that will be used to establish software requirements, defines the policies and procedures used for establishing and controlling software, and describes the SCM system.

1.4 Approach

The SCM discipline is applied to those Configuration Items (CIs) and Computer Software Configuration Items (CSCIs), for which FDC's Major Program Sector has development and/or maintenance responsibilities for on the MRDOC Exhibits. The SCM organization implements the processes described within this plan to ensure that products developed are correct, consistent, complete, and compliant with governing policies.

1.5 System Overview

Software Configuration Management (SCM) is a standard practice at FDC. This SCM Plan will provide an explanation on how FDC's methods and procedures will be used to meet the Configuration Management (CM) requirements of the MRDOC Program.

This plan describes FDC's SCM organization and how this organization will operate to satisfy the MRDOC Program's contractual obligations. FDC's internal CM control methodologies, which are in full compliance with NASA CM implementing directives, will be utilized and modified where required, to support the Microgravity Science Division (MSD) and International Space Station (ISS) Program Configuration Management Requirements. Refer to Figure 1 for an overview of SCM functions in support of the MRDOC Program.

The objectives of this plan are twofold:

To define the FDC SCM system for the MRDOC Program.

To ensure the development and production of the contracted product through maintenance of the product baseline and to prevent unapproved/unauthorized changes to the baseline.

Adherence to this plan will ensure the integrity of the design at all stages of software development and production.

The major sections of this Software Configuration Management Plan are summarized as follows:

ORGANIZATION – The Configuration Manager in the role of Program Configuration Manager (PCM), provides the services to interface and coordinate the support of FDC's Configuration Management Group.

CONFIGURATION IDENTIFICATION – The concept of baseline management is invoked on the design to provide production identification and establish major milestones and points of departure throughout the MRDOC Program life cycle. The major milestones serve as Configuration Management reference points for controlling technical documentation and software configuration.

CONFIGURATION CONTROL – Formal control will be applied to the baselines and their supporting documentation. Changes will be prepared and processed as Engineering Change Orders (ECO), Engineering Change Proposals (ECP), Software Change Orders (SCO) and Software Change Notices (SCN). As the focal point for change management, the Software Configuration Control Board (SCCB) ensures that all changes to released software documents are properly evaluated and approved. Included in this review is the determination of classification for Requests for Deviation/Waiver (RDW).

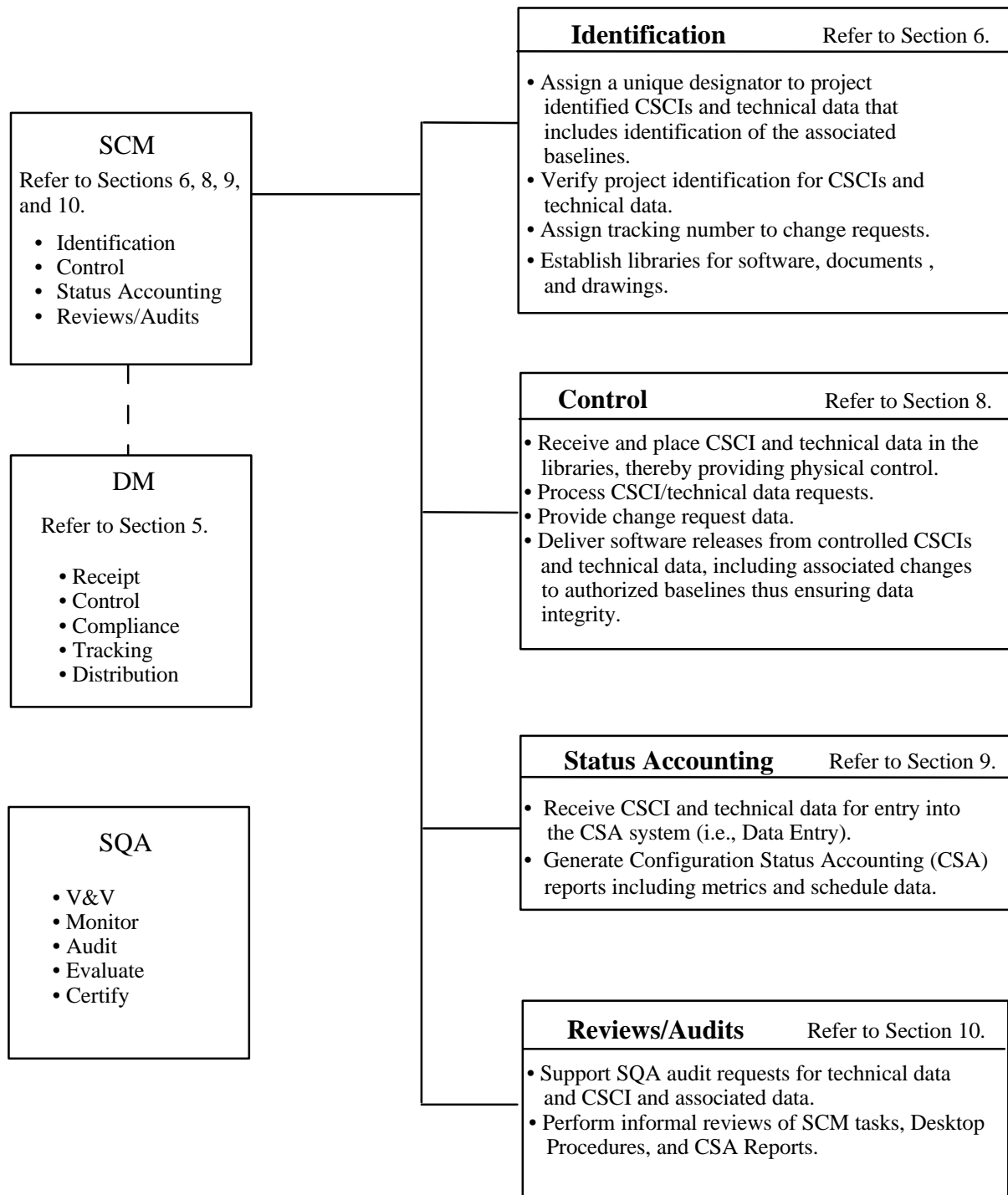


FIGURE 1 Overview of SCM Functions

SOFTWARE CONFIGURATION CONTROL – The disciplines of configuration control will be applied to software throughout its development, maintenance and implementation. Software will be controlled to meet the requirements defined in the Software Requirements Documents (SRD) utilizing a computerized CM tool.

CONFIGURATION STATUS ACCOUNTING – FDC will maintain a complete Product Baseline. Included in these reports and records will be a history of all change activity as identified by approved ECOs, ECPs, SCOs, SCNs and RDWs. Delivered and returned software will be tracked from its original shipped condition to the implementation of any rework, retrofit or revision activity that may be necessary.

INTERFACE MANAGEMENT – FDC will implement effective interface management to ensure smooth integration of all MRDOC Program systems and to ensure interchangeability requirements are achieved where applicable. The steps taken to provide Interface Management are described in section 7.

CONFIGURATION AUDITS – Continual monitoring of engineering and manufacturing documentation will occur throughout the project to insure incremental verification of records. The final As-Built List (ABL), listing all applicable flight software along with their respective versions, will be audited prior to shipment of deliverable product. Basic guidelines and requirements for the contracted configuration audits are contained in section 10 of this text.

SUBCONTRACTOR and VENDOR CONFIGURATION MANAGEMENT – CM will be imposed upon subcontractors and vendors commensurate with the complexity of the software being provided. Vendor and subcontractor documentation and change notification is accomplished by utilizing release notifications and Purchase Requisition (PR) changes to the Purchase Order (PO). Section 11 of this CM Plan outlines CM control procedures for FDC's subcontractors and vendors.

1.6 Order of Precedence for Verification Requirements.

The verification requirements contained in this document shall take precedence over any conflicting verification requirements.

1.7 Acronyms

A list of commonly used acronyms and abbreviations, which have been derived from policies and procedures within this SCM Plan, are presented in Appendix A.

1.8 Definitions

The definitions of applicable terms to this plan are presented in Appendix B.

2.0 DOCUMENTS

This section lists specifications, models, standards, guidelines, handbooks, and other special publications. These documents have been grouped into two categories: applicable documents and reference documents.

2.1 ORDER OF PRECEDENCE FOR DOCUMENTS.

In the event of a conflict between this document and other documents referenced herein, the requirements of this document shall apply. In the event of a conflict between this document and the contract, the contractual requirements shall take precedence over this document. All documents used, applicable or referenced, are to be the issues defined in the Configuration Management (CM) contract baseline. All document changes, issued after baseline establishment, shall be reviewed for impact on scope of work. If a change to an applicable document is determined to be effective, and contractually approved for implementation, the revision status will be updated in the CM contract baseline. The contract revision status of all applicable documents is available by accessing the CM database. Nothing in this document supersedes applicable laws and regulations unless a specific exemption has been obtained.

2.2 APPLICABLE DOCUMENTS.

The documents in these paragraphs are applicable to the MRDOC Program to the extent specified herein.

Document Number	Document Title
EIA/IS-649	National Consensus Standard for Configuration Management
FCF-DOC-002	Science Requirements Envelope Document (SRED) – Fluids and Combustion Facility
FCF-DOC-0058	Problem Reporting and Corrective Action System User's Guide
FCF-ICD-0076	Software Interface Control Document
FCF-LST-0085	Applicable Standards List
MRD-PLN-0002	MRDOC Configuration Management Plan
FCF-PLN-0042	MRDOC Product Assurance Plan
FCF-PLN-0049	Software Assurance Plan
FCF-PLN-0051	Software Management and Development Plan
FCF-SPC-0001	ISS FCF Systems Specification
FCF-SPC-0002	Combustion Integrated Rack (CIR) System Specification
FCF-SPC-0003	Fluids Integrated Rack (FIR) System Specification
FCF-SPC-0004	Shared Accommodation Rack (SAR) System Specification
FCF-SPC-0005	Ground Segment (GS) System Specification
Various (TBD)	Software Design Documents (Multiple element software design documents.)

Document Number	Document Title
Various (TBD)	Software Requirements Documents (Multiple element software requirements documents.)
ISO 10007	Quality Management – Guidelines for Configuration Management
ISO 12207	Information Technology - Quality Management System
MIL-STD-973	Military Standard: Configuration Management
MIL-STD-1521	Audits
MIL-DTL-31000	Technical Data Packages, General Specification For
NASA-STD-2100-91	NASA Software Documentation Standard
P5003	FDC Problem Reporting and Corrective Action (PRACA) System

2.3 REFERENCE DOCUMENTS.

The documents in this paragraph are provided only as reference material for background information and are not imposed as requirements.

Document Number	Document Title
ANSI/IEEE Std 1042-198	IEEE Guide to Software Configuration Management
FCF-DOC-003	Baseline Concept Description
FCF-DOC-004	FCF Concept Compliance Matrix (CCM)
FCF-DOC-005	FCF/CIR Baseline Concept Description
FCF-DOC-006	FCF/FIR Baseline Concept Description
FCF-DOC-007	FCF/SAR Baseline Concept Description
MIL-STD-12	Abbreviations for use on Drawings, and in Specifications, Standard and Technical Documents
PAI 310	Product Assurance Instruction: Configuration Identification and Control
PAI 311	Product Assurance Instruction: Configuration Identification Sheets
PAI 314	Product Assurance Instruction: Deviation/Waiver Request Processing
PAI 315	Product Assurance Instruction: Logbooks and History Sheets
PAI 330	Product Assurance Instruction: Test Procedures
PAI 360	Product Assurance Instruction: Non-Conformance Identification & Disposition
PAI 440	Product Assurance Instruction: Problem Reporting, Analysis, & Corrective Action System (PRACA)
PAI 500	Product Assurance Instruction: Software Assurance Policy

Document Number	Document Title
PAI 510	Product Assurance Instruction: Software Configuration Management
PAI 520	Product Assurance Instruction: Software Reviews, Audits & Surveys
PAI 540	Product Assurance Instruction: Software Documentation Reviews
PAI 550	Product Assurance Instruction: Software Testing
SARGE	Standard Assurance Requirements and Guidelines for Experiments (Revision "0")
SSP 50007	Space Station Inventory Management System Label Specification
SSP-50200	Station Implementation Plan
SSP 57000	Pressurized Payloads Interface Requirements Document

2.4 SOURCE OF DOCUMENTS

Copies of the referenced documents shall be available from MRDOC Program Library.

2.5 TAILORING

FDC tailors its Configuration Management of software to the needs of the contract, resulting in an effective, yet cost efficient program. This plan reflects FDC's current SCM program for the MRDOC Program. Should additions, revisions or enhancements to the SCM requirements be necessary, this SCM Plan may be modified accordingly. Requests to amend this existing program must be made in writing through the appropriate contractual channels. All amendments will require formal review and approval, and will result in a document revision.

3.0 SOFTWARE CONFIGURATION MANAGEMENT

This section describes the SCM organization in relation to the program and project organization structure.

3.1 SCM ORGANIZATION

The organizational structure for the MRDOC Program is an extension of the overall management structure at Federal Data Corporation, Major Programs Sector as shown in Figure 2. Each principal operation within the Major Programs Sector is supported by personnel who report to the MRDOC Program Manager. The flow down of responsibility ensures the dissemination of company policy and procedures throughout the entire operation. The MRDOC Program Manager's responsibility extends across all functions. To ensure that uniform and effective CM disciplines are maintained, the Configuration Manager ensures that the policies are consistent and not compromised by individual projects. The Program Configuration Manager (PCM) is responsible for configuration management policies and direction and the implementation of the configuration management functions for FDC's MRDOC Program. Hardware and software configuration management operations are combined under a single CM department as shown in Figure 3.

3.2 SCM RESPONSIBILITIES

SCM is responsible for maintaining configuration control over software Developmental Configurations and Baselines, and for processing changes to the software configuration. SCM functions include Software Development Library (SDL) operation, software product release coordination, and change request processing and tracking.

The responsibilities of each SCM function are listed in the paragraphs below.

3.2.1 Configuration Identification.

- a. Establish methods and procedures for unique identification of CSCIs.
- b. Establish and maintain Functional, Allocated, and Product Baselines and the Developmental Configuration (identify, document, archive, and track changes to system releases).
- c. Establish and follow release procedures to obtain Product Baselines for new version releases.
- d. Coordinate assignment of identifying numbers for CSCIs and documents.
- e. Provide documentation that reflects the released software package.
- f. Coordinate release of software and associated documentation to release organizations.
- g. Maintain records and prepare reports on release coordination activities.

Software Configuration Management Interface within the MRDOC Organization

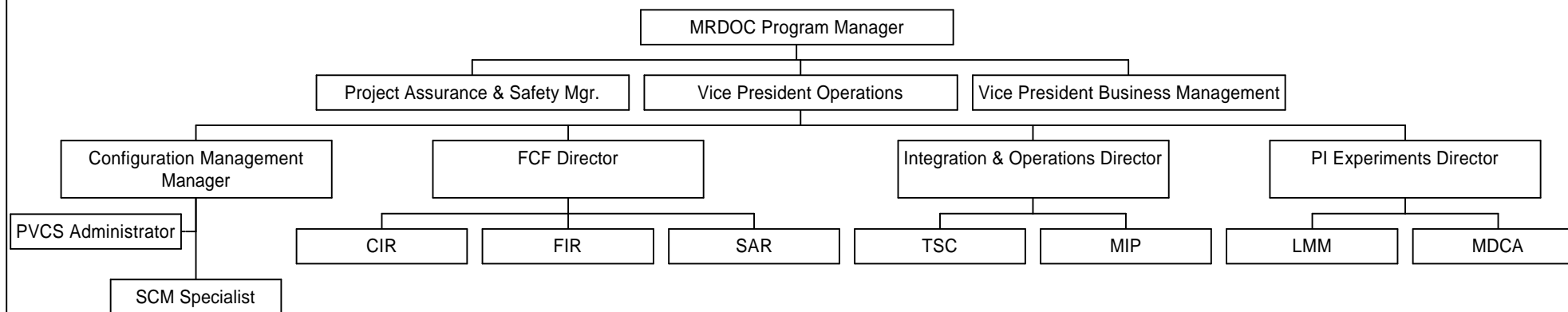


Figure 2

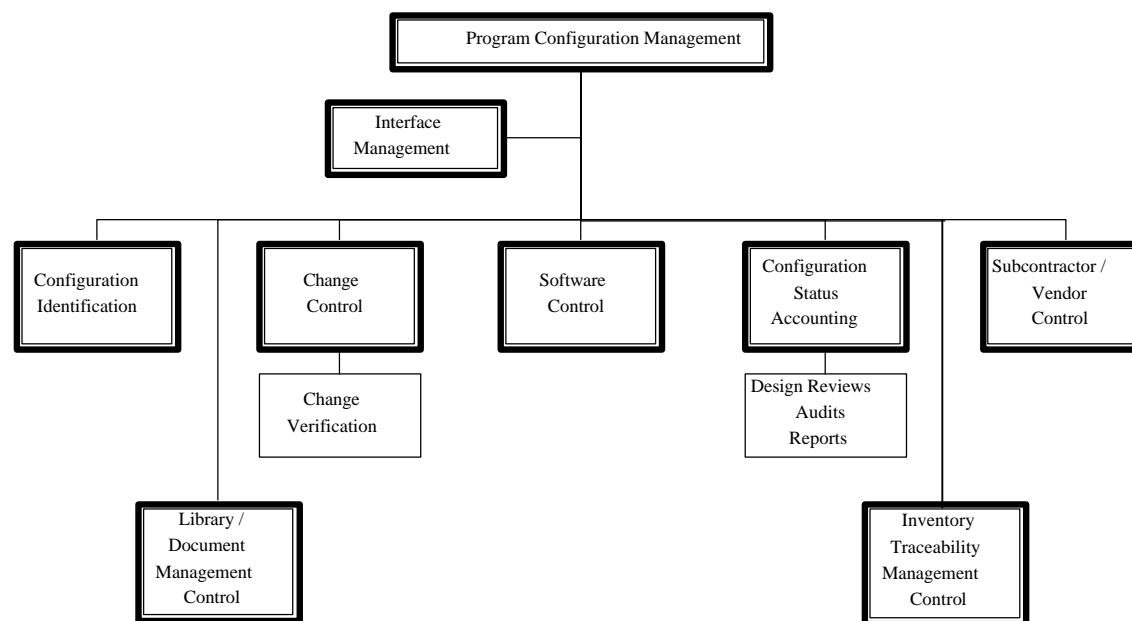


Figure 3 Configuration Management Organizational Structure

3.2.2 Configuration Control

- a. Serve as a member of the Software Configuration Control Board (SCCB). SCM is responsible for preparing and distributing the meeting agenda and minutes and for recording action items and their resolution.
- b. Establish and document configuration change control procedures.
- c. Establish and follow configuration controls for software and documentation.
- d. Place contents of baseline and Developmental Configurations under configuration control in the SDL.
- e. Generate executable load modules from controlled source code.
- f. Ensure that the contents of the SDL are changed by SCM personnel and only upon receipt of the appropriate paper work signed by the Configuration Manager.
- g. Prepare and maintain master(s) of the currently active version of each CI until superseded by a new version. Retain superseded versions of the master(s) in the SDL archive files.
- h. Maintain records and prepare reports on SDL activities and software products.
- i. Perform non-technical check of software documentation.

3.2.3 Configuration Status Accounting (CSA)

- a. Provide CSA recording and reporting.
- b. Maintain accounting of software changes by tracking change requests, ensuring traceability to a formal change proposal (i.e., ECP) from initiation through resolution and disposition.
- c. Prepare status reports on change requests, formal change proposals (i.e., ECPs), and changes.

3.2.4 Configuration Audits

- a. Support requests for audit and certification of software systems by SQA or the independent auditor.
- b. Perform reviews of SCM activities and products.
- c. Review and update SCM documentation as required to ensure that current applicability is maintained.

3.2.5 Training

The Configuration Manager is responsible for identifying, establishing, coordinating, and revising training as required to ensure effective performance of SCM activity by the SCM organization and software-related groups.

3.2.6 Program Configuration Manager

The Configuration Manager assumes the responsibilities of Program Configuration Manager (PCM). As PCM he provides the administrative and technical direction required to execute configuration controls in accordance with the program contract requirements. This position reports directly to the Vice President Operations and serves as Co-Chairperson of the SCCB. This position provides program configuration management interface between the MRDOC group, Major Subcontractors and NASA's organizations.

3.2.7 SCM Project Specialist

Reporting to the Configuration Manager, SCM Project Specialists are assigned the responsibility of interfacing with the functional organizations within FDC for coordinating the accomplishment of these tasks:

- Establish and implement Program specific SCM operating policies and procedures for CM activities for the MRDOC Program.
- Prepare, maintain, and implement the MRDOC Software Configuration Management Plan.
- Support PCM and FDC SCCB activities.
- Create and maintain engineering Drawing and Document Number Logs.
- Process ECO's/ECP's, PIRN's, Change Notices (CN's), RDW's, and other related documentation supporting software development, test, and integration activities.
- Coordinate FDC evaluation and impact assessment for changes proposed by FDC, NASA, or other contractors.
- Provide all aspects of CM control pertaining to audit, status accounting, software control, and subcontractor interface.
- Maintain engineering change status, and release, and implementation verification status.
- Establish and maintain a MRDOC Project Library.
- Maintain the interface between Property Management and Inventory Control Systems for the control and traceability of software designated for the MRDOC Program.
- Assist in conducting any required configuration inspections of hardware/software prior to acceptance and support of Acceptance Reviews.

3.2.8 Library Project Control Specialist

The Library Project Control Specialist will:

- Update the text documents to incorporate Engineering Change Orders (ECO's) and Engineering Change Proposals (ECP's).
- Maintain controlled distribution lists.
- Distribute ECO's, ECP's, and RDW's.
- Maintain masters of all released files including software (hardcopy and electronic version as required).
- Control and maintain the MRDOC CM Web site. Program related information and documentation is maintained for team members' use.

3.2.9 Software Administrator.

The Software Administrator will:

- Configure and administer automated cm tools, MS SQL Server, MS ISS Server, and Software Repository.
- Perform Software Configuration Management (SCM) audits.
- Support SCM activities.
- Assist FDC in achieving Capability Maturity Model (CMM), Level III.

3.3 PROJECT ORGANIZATIONAL STRUCTURE

Figure 4 is a graphic representation of the program and project organizational structure with respect to the SCM organization. Although SCM takes direction from Program Operations Management, it operates within the policies and procedures established by the MRDOC Program Configuration Management Office. Listed below are the responsibilities of each of the organizations as related to MRDOC Program software development.

SCM interfaces with the functions listed below to control software configuration and release activities.

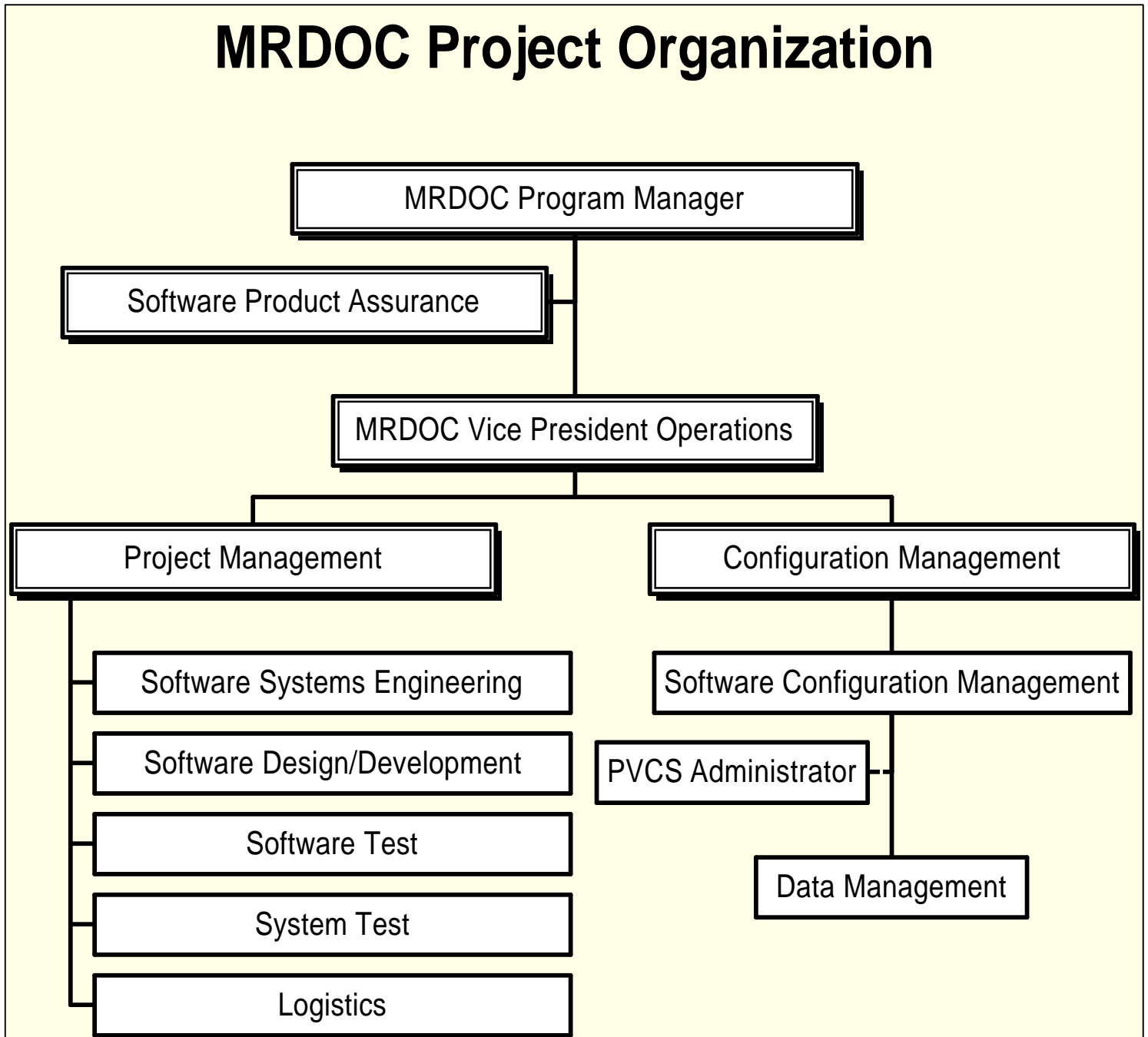


FIGURE 4. MRDOC Project/SCM Organization Structure

- **Program Management** - Responsible for and has the authority to ensure complete fulfillment of all program requirements. The MRDOC Program Manager has the overall responsibility for acquisition, funding, and transitioning of the various exhibits and projects.
- **Project Management** - Responsible for the technical aspects of the project. The Project Manager has the responsibility for local funding, allocations, scheduling, tasking, and reporting to program management.

- **Software Systems Engineering** - Responsible for systems design (and associated documentation) overview and guidance; detailed design and coding; test plans, procedures, and reports; software unit testing; and preliminary CSCI testing.
- **Software Design and Development** - Responsible for software design (and associated documentation) overview and guidance; detailed design and coding; test plans, procedures, and reports; software unit testing; and preliminary CSCI testing.
- **Software Test** - Responsible for the conduct of software testing, including preparation of test plan, description, procedures, and reports. Software Test Group ensures that the correct configuration is undergoing test and incorporates approved changes into released test documentation based on change request baselining data from SCM. Software Test Group confirms verification of change request corrective measures prior to change request closure. SCM identifies all change requests included in an Engineering Master (EM) that is to be tested. Test personnel then provide SCM a copy of the test report.
- **Software Quality Assurance (SQA)** - Responsible for auditing the software development activities and products (FCA and PCA) and certifying of SCM compliance with this plan and desktop procedures (DTPs).
- **System Test** - Responsible for administering the verification and validation (V&V) testing prior to release of the software. The System Test Group is a separate organization from the Software Development Group (i.e., the Software Systems Engineering Group and the Software Design and Development Group).
- **Logistics** - Responsible for ensuring that changes made to a system are supportable. SCM provides CSCI and associated technical data for logistics evaluation.
- **Data Management (DM)** - Responsible for the receipt, distribution, and tracking of technical data associated with the project. DM also ensures compliance with contract requirements as defined in the Contract Data Requirements List (CDRL).

3.3.1 Control/Review Board

The Software Configuration Control Board is responsible for overseeing required configuration control for all software and its supporting documentation on the MRDOC Program. The board controls the configuration of Flight Code, Ground Code, Utilities, and Computer Software Executable Images (CSEI) as well as associated documentation and ancillary support software from the initial release of design documentation to the delivery of software. The MRDOC SCCB will review customer-initiated changes (including new/revised document releases) to assess feasibility, possible impacts, (cost, schedule, etc.) and provide a final recommendation. The SCCB will review all FDC initiated change order requests for validity and to determine the proper classification. Deviations and waivers will also be reviewed by the SCCB to determine classification as major or minor. The SCCB is chaired by the MRDOC Vice President

Operations with the PCM as co-chairperson and secretary. The following disciplines have representation on the SCCB:

- Software Engineering
- Test Engineering
- Product Assurance
- Systems Engineering
- Electrical Engineering
- Mechanical Engineering
- Optics Engineering
- Safety/Reliability/Maintainability
- Configuration Management
- Integrated Logistics Support
- Ad Hoc Members

The SCCB member relationships are shown in Figure 5. Member responsibilities and overall actions are further described in section 8. SCM provides status accounting reports to the SCCB and updates the status accounting database to reflect SCCB decisions.

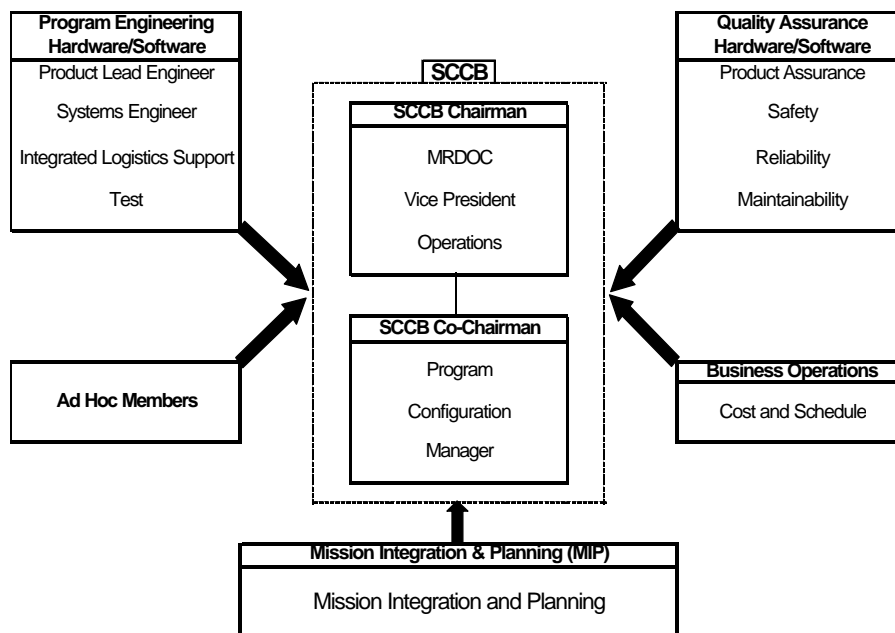


FIGURE 5. MRDOC Software Configuration Control Board

3.4 RELATIONSHIP OF CM TO THE SOFTWARE PROCESS LIFE CYCLE

The principles of configuration management will be formally and informally applied to MRDOC software and all associated documentation throughout the entire life of the software process. During the initial phases of development, formal control will pertain to approved documentation released for the functional, allocated and developmental baselines. Version control for preliminary and developmental unit software will be maintained in an electronic repository where historical data will be tracked and retained by the Software Administrator and the software organization. Upon maturity of the unit modules, to where the unit software development is functional and complete (just prior to the Functional Configuration Audit), the Product Baseline will be established and formal configuration control will be implemented for all activities from this point forward.

The Configuration Management group will manage all software related engineering data pertaining to identification, change control, release, status accounting and reporting, audit/verification and repository control functions depicted in Figure 6. Data Management for the MRDOC Program is further defined in section 5.

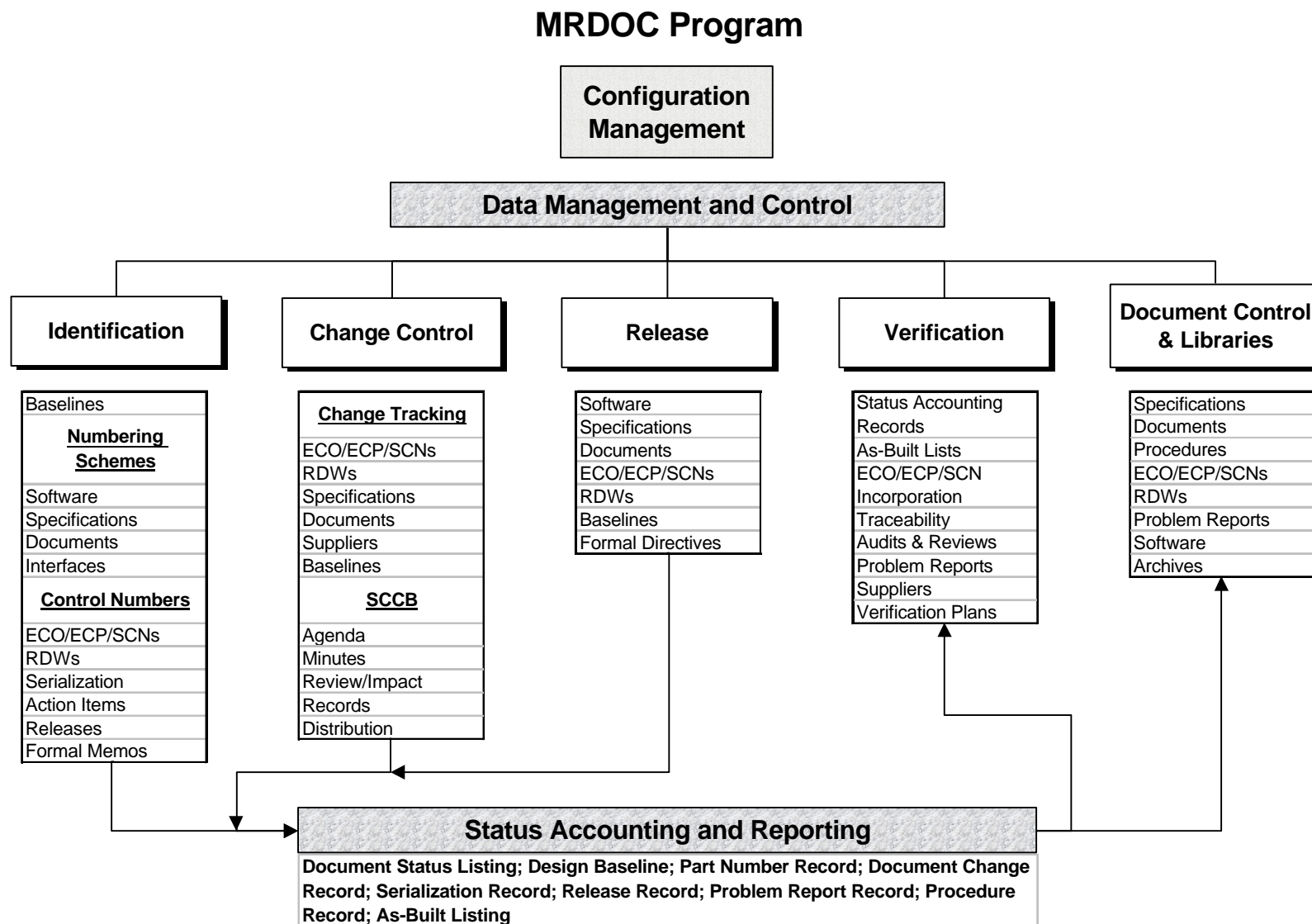


Figure 6 Configuration Data Management

4.0 CM PHASING AND MILESTONES

Milestones serve as Configuration Management reference points for controlling technical documentation and software configuration. The concept of baseline management establishes major milestones and points of departure throughout the life cycle of the MRDOC Program. Coupled with the formal reviews and audits defined in section 10, progressive accomplishments tied to the project's software development process can be profiled. The following is a summary of MRDOC Program software activity milestones:

- The Software Specification Review and the establishment of the Allocated Baseline indicate the start of full-scale development and preliminary design efforts.
- The Preliminary Design Review signifies CSCI informal developmental configuration and the initiation of detailed design activities.
- The Critical Design Review indicates that 90% of the flight documentation as well as engineering modeling are complete. Unit coding and testing and CSCI integration and testing can now be accomplished.
- The Verification and Testing Review enacts the overall integration and test plan and permits CSCI qualification testing.
- The Functional Configuration Audit commences product baseline activities (formal CM control) and signifies the completion of development efforts and the beginning of hardware/software integration and testing.
- The Physical Configuration Audit results in establishment of the Product Baseline for the hardware CIs and permits the progression to system level qualification testing, Principal Investigator characterization, and project simulation.

This section describes the software development activity for software-related groups and the SCM responsibilities in relation to this activity and program events. These activities occur within the phases of the software life cycle. The software life cycle includes five phases: Requirements Definition, Design, Development, Installation and Deployment, and Operations and Support. Some of the activities may be applicable to and overlap with other life cycle phases. For this reason, the objectives of each life cycle phase are presented. Table 4-1 defines the SCM milestones in relation to software-related group activity for the MRDOC Program.

4.1 REQUIREMENTS DEFINITION

The objectives of the Requirements Definition phase are to:

- a. Explore various material alternatives to satisfying the documented mission need.
- b. Define the most promising system concept(s).
- c. Develop supporting analyses and information to include identifying high-risk areas and risk management approaches to support project decisions.
- d. Develop a proposed acquisition strategy and initial program objectives for cost, schedule, and performance for the most promising system concept(s).

SOFTWARE-RELATED GROUP ACTIVITY	SCM MILESTONE
Concept	Plan how project will affect program products and their CM Baseline product identification
Project Planning and Oversight	Draft SCMP (new system) or SCMP update (existing system) SCM organization established, staffed Management and technical review participation Configuration identification Planning documents under configuration control Establish SCCB
Software Development Environment	SCM staff training SDL and SDFs established
System Requirements Analysis	System requirements documents under configuration control SCCB support
System Design	Approved SCMP implemented SCM tasks identified DTPs created and/or maintained System design documents baselined and maintained Functional Baseline established and maintained CSA system established and maintained CM Document Library established and maintained
Software Requirements Analysis	Software requirements documents baselined Allocated Baseline established CM Drawing Library established and maintained
Software Design	Development Configuration products maintained Development Configuration corrective action process established
Software Implementation and Unit Testing	Baseline software development folders and associated artifacts
Unit Integration and Testing	Baseline integration test plans, procedures and results
CSCI Qualification Testing	Test documents baselined
CSCI/HWCI Integration and Testing	FCA and PCA support
System Qualification Testing	Baseline Test Reports
Software Use Preparation	Product Baseline established and maintained Software user documents and manuals baselined
Software Transition Preparation	Product Baseline archived Product Baseline transferred to SSA

FIGURE 7. SCM Milestones

SCM responsibilities are to:

- a. Develop a CM Plan for the project, if tasked.
- b. Charter the SCCB.
- c. Document the Functional and Physical Characteristics (FPC).
- d. Ensure control and accounting of the FPC.
- e. Participate in System Requirements Review.

4.2 DESIGN

Objectives of the Design phase are to:

- a. Define critical design characteristics and expected capabilities of system concept(s).
- b. Demonstrate that the technologies critical to the most promising concept(s) can be incorporated into system design(s) with confidence.
- c. Prove that the processes critical to the most promising system concept(s) are understood and attainable.
- d. Develop the analysis/information needed to support project decisions.
- e. Establish a proposed Development Baseline containing refined program cost, schedule, and performance objectives for the most promising design approach.

SCM responsibilities are to:

- a. Update CCB charter and CM Plan.
- b. Continue documentation of the FPC.
- c. Ensure contractor control and accounting of the FPC.
- d. Ensure Government control and accounting of the FPC.
- e. Participate in System Design Review (SDR).

4.3 DEVELOPMENT

Table 4-1 shows SCM milestones for the Development phase of a software life cycle.

4.3.1 Requirements Definition

During requirements definition, software related groups are concerned with the following activities:

- a. Provide sponsors with estimates of cost, schedule, risk items, etc.
- b. Assist with generation of an action plan to include initial estimates for cost, schedule, risk, and system size.
- c. Involve Software Quality Assurance (SQA) in planning.

SCM responsibilities are to:

- a. Plan how this project will affect other program products and the configuration management of them.
- b. Baseline the product identification.

4.3.2 Project Planning and Oversight

During project planning and oversight, software-related groups are concerned with the following activities:

- a. Software development planning. Development and documentation of plans to conduct software development process activities identified in the following sections; development of program and project plans including a SDP and development and implementation of a CM policy.
- b. CSCI test planning. Development and documentation of plans for conducting CSCI qualification testing and the generation of a Software Test Plan (STP).
- c. System test planning. Participation in developing and documenting plans to conduct system qualification testing.
- d. Software installation planning. Development and documentation of plans to perform software installation and training at user sites and generation of a Software Installation Plan (SIP).
- e. Software transition planning. Identification of all software development resources needed by the support agency to fulfill support concept, and development and documentation of a Software Transition Plan (STrP).
- f. Following and updating plans. Conduct of relevant activities in accordance with approved plans, supporting management reviews of the software development process, and updating plans as needed.
- g. Establishment of the SCCB.

SCM responsibilities are to:

- a. Create a draft SCMP or update an SCMP for existing system.
- b. Establish and staff the project SCM functional organization.
- c. Apply and maintain identification scheme for project products.
- d. Place planning documents (SDP, STP, SIP, STrP, SCMP) under configuration control.
- e. Participate in joint management and technical reviews.

4.3.3 Establishment of Software Development Environment

During establishment of a software development environment, software-related groups are concerned with the following activities:

- a. Software engineering environment. Establishment, control, and maintenance of the environment.
- b. Software test environment. Establishment, control and maintenance of the environment.

- c. Software Development Library (SDL). Establish, control, and maintain an SDL to facilitate the orderly development and subsequent support of software.
- d. Software Development Files (SDFs). Establishment, control, and maintenance of an SDF for each software unit or logically related group of software units.
- e. Non-deliverable software. Verification that the non-deliverable software performs the intended functions.

SCM responsibilities are to:

- a. Train staff on the SCM processes.
- b. Establish and maintain the SDL and SDFs.
- c. Participate in joint management and technical reviews.

4.3.4 System Requirements Analysis

During system requirements analysis, software-related groups are concerned with the following activities:

- a. Analysis of user input. Analysis provided by acquirer and generation of need surveys, problem/change reports, feedback on prototypes, interviews, or other user input.
- b. Operational concept. Participation in the definition and documentation of the operational concept for the system and generation of an Operational Concept Description (OCD).
- c. System requirements. Participation in the definition and documentation of system requirements and methods used to ensure that each requirement has been met; and, depending on CDRL provisions, generation of a System/Subsystem Specification (SSS) or an Interface Requirements Specifications (IRSs).
- d. Participation in joint management and technical reviews.

SCM responsibilities are to:

- a. Participate in joint management and technical reviews to provide status on SCM activities.
- b. Place system requirements documents (OCD, SSS, IRSs) under configuration control.
- c. Support the SCCB.

4.3.5 System Design

During system design, software-related groups are concerned with the following activities:

- a. System-wide design decisions. Participation in definition and documentation of system-wide design decisions, and generation of a System/Subsystem Design Description (SSDD),

Interface Design Descriptions (IDDs), or Database Design Descriptions (DBDDs), depending upon CDRL requirements.

- b. Architectural design. Participation in definition and documentation of architectural design and traceability between system components and system requirements.
- c. Convene the SCCB to establish the Functional Baseline.
- d. Participation in joint management and technical reviews.
- e. Approve project plans: Program and Project Plans, Software Development Plan, and SCMP.

SCM responsibilities are to:

- a. Implement approved SCMP.
- b. Identify tasks stated in SCMP.
- c. Participate in joint management and technical reviews.
- d. Place system design documents (SSDD, IDD, DBDD) under configuration control.
- e. Maintain configuration control of the Functional Baseline.
- f. Support the SCCB.
- g. Establish and maintain CSA system.
- h. Provide access procedures to project personnel on use of CSA system.
- i. Generate and distribute CSA reports.
- j. Establish and maintain the CM Document Library.

4.3.6 Software Requirements Analysis

During software requirements analysis, software-related groups are concerned with the following activities:

- a. Software requirements. Participate in the definition and documentation of CSCI software requirements in Software Requirements Specifications (SRSs) or the IRSs, methods used to ensure requirements have been met, and traceability between CSCI requirements and system requirements.
- b. Convene the SCCB to establish the Allocated Baseline.
- c. Participation in joint management and technical reviews.

SCM responsibilities are to:

- a. Place software requirements documents (SRSs, IRSs) under configuration control.
- b. Maintain configuration control of the Functional and Allocated Baselines.
- c. Participate in joint management and technical reviews.
- d. Support the SCCB.
- e. Maintain CSA system.

- f. Generate and distribute CSA reports.
- g. Maintain the CM Document Library.
- h. Establish and maintain the CM Drawing Library.

4.3.7 Software Design

During software design, software-related groups are concerned with the following activities:

- a. CSCI-wide design decisions. Participation in definition and documentation of CSCI-wide design decisions in design documentation.
- b. CSCI architectural design. Participation in definition and documentation of CSCI architectural design in Software Design Documents (SDDs) or IDDs and traceability between software units and CSCI requirements.
- c. CSCI detailed design. Participation in development and documentation of descriptions for each software unit in design documentation.
- d. Convene the SCCB to establish Developmental Configuration.
- e. Participation in joint management and technical reviews.

SCM responsibilities are to:

- a. Establish and maintain corrective action process for Developmental Configuration.
- b. Place SDDs under developmental configuration control.
- c. Maintain configuration control of developmental configuration products.
- d. Maintain configuration control of Functional and Allocated Baselines.
- e. Participate in joint management and technical reviews.
- f. Support the SCCB.
- g. Maintain CSA system and distribute CSA reports.
- h. Maintain CM Document and Drawing Libraries.

4.3.8 Software Implementation and Unit Testing

During software implementation and unit testing, software-related groups are concerned with the following activities:

- a. Software Implementation. Development and documentation of software corresponding to each software unit in the CSCI design.
- b. Preparation for unit testing. Establishment of test cases, test procedures, and test data for testing the software corresponding to each software unit, and documentation of test case information in SDFs.

- c. Performance of unit testing. Testing the software corresponding to each software unit in accordance with unit test cases and procedures.
- d. Revision and retesting. Software revision, retesting, and SDF update based on unit testing results.
- e. Analyzing and recording unit testing results. Analyzing unit testing results and documentation of test and analysis results in appropriate SDFs.
- f. Participation in joint management and technical reviews.

SCM responsibilities are to:

- a. Maintain corrective action process and provide status reports.
- b. Maintain configuration control of developmental configuration products (including source code and source code listings).
- c. Maintain configuration control of the Functional and Allocated Baselines.
- d. Participate in joint management and technical reviews.
- e. Support the SCCB.
- f. Maintain CSA system and distribute CSA reports.
- g. Maintain the CM Document and Drawing Libraries.
- h. Maintain the SDL and SDFs.

4.3.9 Unit Integration and Testing

During unit integration and testing, software-related groups are concerned with the following activities:

- a. Preparation for unit integration and testing. Establishment of test cases, test procedures, and test data to conduct unit integration and testing, and documentation of information in appropriate SDFs.
- b. Performance of unit integration and testing. Performance of unit integration and test in accordance with unit integration test cases and procedures.
- c. Revision and retesting. Revision of software, retesting, and updating of SDFs and other software products based on results of unit integration and testing.
- d. Analysis and recording unit integration and test results. Analysis of unit integration and testing results and documentation of these results in appropriate SDFs.
- e. Participation in joint management and technical reviews.

SCM responsibilities are to:

- a. Maintain corrective action process and provide status reports.
- b. Maintain configuration control of developmental configuration products.
- c. Maintain configuration control of the Functional and Allocated Baselines.

- d. Participate in joint management and technical reviews.
- e. Support the SCCB.
- f. Maintain the CSA system and distribute CSA reports.
- g. Maintain the CM Document and Drawing Libraries.
- h. Maintain the SDL and SDFs.

4.3.10 CSCI Qualification Testing

During CSCI qualification testing, software-related groups are concerned with the following activities:

- a. Independence in CSCI qualification testing. Assurance that qualification testing is performed by nonparticipant in the CSCI detailed design and implementation.
- b. Testing on target computer system. Inclusion of CSCI qualification testing on target computer system or approved alternative system.
- c. Preparation for CSCI qualification testing. Definition and documentation of test preparations, cases, and procedures for CSCI qualification testing, traceability between test cases and the CSCI requirements, and generation of a Software Test Description (STD).
- d. Dry run of CSCI qualification testing. Testing in preparation for witnessing by the acquirer; documentation of results in SDFs and update of CSCI test cases and procedures.
- e. CSCI qualification testing. Performance of CSCI qualification testing in accordance with the CSCI test cases and procedures.
- f. Revision and retesting. Revision of software, conduct of all necessary retesting, and update of SDFs and other software products, based on results of CSCI qualification testing.
- g. Analysis and recording of CSCI qualification test results. Analysis and documentation of test results in a Software Test Report (STR).
- h. Participation in joint management and technical reviews.

SCM responsibilities are to:

- a. Maintain corrective action process and provide status reports.
- b. Place testing documents (STD, STR) under developmental configuration control.
- c. Maintain configuration control of developmental configuration products.
- d. Maintain configuration control of the Functional and Allocated Baselines.
- e. Participate in joint management and technical reviews.
- f. Support the SCCB.
- g. Maintain CSA system and distribute CSA reports.
- h. Maintain the CM Document and Drawing Libraries.
- i. Maintain the SDL and SDFs.

4.3.11 CSCI/Hardware Configuration Item (HWCI) Integration and Testing

During CSCI/HWCI integration and testing, software-related groups are concerned with the following activities:

- a. Preparation for CSCI/HWCI integration and testing. Participation in development and documentation of test cases, test procedures, and test data for conduct of CSCI/HWCI integration and testing, and documentation of software-related information in the appropriate SDFs.
- b. Performance of CSCI/HWCI integration and testing. Participation in CSCI/HWCI integration and testing in accordance with the CSCI/HWCI integration test cases and procedures.
- c. Revision and retesting. Revisions to software, participation in all necessary retesting, and update of appropriate SDFs and other software products, based on CSCI/HWCI integration and testing results.
- d. Analysis and recording CSCI/HWCI integration and test results. Participation in analysis of CSCI/HWCI integration and testing results, and documentation in appropriate SDFs.
- e. Participation in joint management and technical reviews.
- f. Conduct of FCA and PCA.

SCM responsibilities are to:

- a. Maintain corrective action process and provide status reports.
- b. Maintain configuration control of developmental configuration products.
- c. Maintain configuration control of the Functional and Allocated Baselines.
- d. Participate in joint management and technical reviews.
- e. Support FCA and PCA.
- f. Support the SCCB.
- g. Maintain CSA system and distribute CSA reports.
- h. Maintain the CM Document and Drawing Libraries.
- i. Maintain the SDL and SDFs.

4.3.12 System Qualification Testing

During system qualification testing, software-related groups are concerned with the following activities:

- a. Independence in system qualification testing. Insurance that system qualification testing is performed by nonparticipant in the detailed design and implementation of system software.

- b. Testing on target computer system. Qualification testing on target computer system or approved alternative system.
- c. Preparation for system qualification testing. Participation in development and documentation of test preparations, test cases, and test procedures to be used for system qualification testing, traceability between test cases and system requirements, and documentation of all applicable items in the STD.
- d. Dry run of system qualification testing. Testing in preparation for witnessing by the acquirer, documentation of results in SDFs and update of system test cases and procedures.
- e. Performance of system qualification testing. Participation in system qualification testing in accordance with the system test cases and procedures.
- f. Revision and retesting. Participation in all software revision, retesting, and update of appropriate SDFs and other software products, based on results of system qualification testing.
- g. Analysis and recording of system qualification test results. Participation in analysis and documentation of system qualification test results.
- h. Participation in joint management and technical reviews.

SCM responsibilities are to:

- a. Maintain corrective action process and provide status reports.
- b. Maintain configuration control of Functional and Allocated Baselines.
- c. Participate in joint management and technical reviews.
- d. Support the SCCB.
- e. Maintain CSA system and distribute CSA reports.
- f. Maintain the CM Document and Drawing Libraries.
- g. Maintain SDL and SDFs.

4.3.13 Software Use Preparation

During software use preparation, software-related groups are concerned with the following activities:

- a. Preparation of executable software. Preparation of executable software for each user site and documentation of all applicable items in the Software Product Specification (SPS).
- b. Preparation of version descriptions for user sites. Identify and document the exact version of software prepared for each user site in a software Version Description Document (VDD).
- c. Preparation of user manuals. User manuals may include System User's Manual (SUM), Software Input/Output Manual (SIOM), Software Center Operator Manual (SCOM), and Computer Operation Manual (COM).
- d. Installation at user sites. Installation, check out of executable software at specified user sites, training, and other specified assistance.

- e. Convene the SCCB to establish the Product Baseline.

SCM responsibilities are to:

- a. Place software user documents (SPS, VDD) and user manuals (SUM, SIOM, SCOM, COM) under configuration control.
- b. Maintain corrective action process and provide status reports.
- c. Maintain configuration control of Functional, Allocated and Product Baselines.
- d. Participate in joint management and technical reviews.
- e. Support the SCCB.
- f. Maintain CSA system and distribute CSA reports.
- g. Maintain the CM Document and Drawing Libraries.
- h. Maintain SDL and SDFs.

4.3.14 Software Transition Preparation

During software transition preparation, software-related groups are concerned with the following activities:

- a. Preparation of executable software. Preparation of executable software for transition to support site and documentation of applicable items in the SPS.
- b. Preparation of source files. Preparation of source files for transition to the support site and documentation of applicable items in the SPS.
- c. Preparation of version descriptions for support site. Identification and documentation of the exact version of software prepared for the support site in the VDD.
- d. Preparation of the "as-built" CSCI design and related information. Update of each CSCI design description to match the "as-built" software. Definition and documentation of all information (in the SPS) needed to support the software, and traceability between the CSCI's source files and software units and between the computer hardware resource utilization measurements and the CSCI requirements concerning them.
- e. Update of system design description. Participation in updating system design description to match the "as-built" system in the SSDD.
- f. Preparation of support manuals. Support manuals may include Computer Programming Manuals (CPMs) and Firmware Support Manuals (FSMs).
- g. Transition to designated support site. Installation and check out of deliverable software in the support environment, training and miscellaneous assistance to support agency.

SCM responsibilities are to:

- a. Archive Product Baseline.
- b. Transfer Product Baseline to support site.

4.4 INSTALLATION AND DEPLOYMENT

Objectives of the Installation and Deployment phase of the software life cycle are to:

- a. Establish a stable, efficient production and support base.
- b. Achieve an operational capability that satisfies the mission need.
- c. Conduct follow-on operational and production verification testing to confirm and monitor performance and quality and verify the correction of deficiencies.

SCM responsibilities are to:

- a. Update CCB charter, CM Plan(s), Functional, Allocated and Product Baselines.
- b. Ensure contractor and government control of FPC, Functional, Allocated, and Product Baselines.
- c. Provide training in the CM process to the operating forces.

4.5 OPERATIONS AND SUPPORT

Objectives of the Operations and Support phase of the software life cycle are to:

- a. Ensure that the fielded system continues to provide the capabilities required to meet the identified mission need.
- b. Identify shortcomings or deficiencies that must be corrected to improve performance.

SCM responsibilities are to:

- a. Update CCB charter, CM Plan(s), Functional, Allocated and Product Baselines.
- b. Continue control and accounting of FPC, Functional, Allocated, and Product Baselines.
- c. Participate in conduct of audits as required.
- d. Provide training in the CM process to the operating forces.

5.0 DATA MANAGEMENT

This section describes the methods for meeting the configuration management technical data requirements of the MRDOC Program. The section describes the data handling, processing, storage, integrity, transfer, security, and maintenance of configuration management technical data.

FDC has established a formal working library to ensure a repository of technical data is maintained for the program. This section describes the methods for data handling, processing, storage, integrity, transfer, security, and maintenance of CM technical data.

Data management responsibilities are:

- a. Receive/obtain CDRL documents, software, or project technical data.
- b. Implement and apply the configuration identification scheme in accordance with Section 6 of this plan.
- c. Catalogue the CDRL documents, software, or project technical data.
- d. Maintain status records or database of CDRL documents, software, or project technical data.
- e. Perform security access and control.
- f. Provide change control.
- g. Provide distribution copies for project personnel or for outside distribution.
- h. Maintain review comments or files, and forward comments to document originators.
- i. Prepare and distribute status and inventory reports.
- j. Archive CDRL documents, software, or project technical data.
- k. Track CDRL documents, software, or project technical data requiring response or action.

5.1 DATA DISTRIBUTION AND ACCESS

Access to data is limited in accordance with the applicable distribution statements defined by the contract and by data rights, CDRL distribution, security requirements, and data status level (released or submitted for approval unless otherwise specified). Distribution statements are included as part of the electronic medium or physically affixed on data items. Distribution statements are in accordance with the contract.

Access to project technical data is limited in accordance with the applicable distribution statements defined by the contract or Project Manager and by data rights, CDRL distribution security requirements, and data status level (released or submitted for approval unless otherwise specified). Distribution lists of technical data are maintained as part of the data status reporting function. Request for project technical data by activities outside of this project require approval by the Project Manager or designated authority.

5.2 AUTOMATED PROCESSING AND SUBMITTAL OF DATA

The following requirements are used to identify and control data during the review and update cycle:

- a. Data files are uniquely identified and include file version and "submitted" status, e.g., "working", "released", etc. File naming conventions are used to indicate changes from previous versions or to distinguish an altered (annotated, redlined) file version from the originally submitted file version (e.g., filename.srs;2, or filename_srs.ann;6).
- b. Data and changes are transmitted in accordance with the submittal date specified by the contract.
- c. An acknowledgment of receipt from the receiving party is required when electronic data is being sent to the Customer. The required time to respond is 24 hours. A follow-up is made after the 24-hour period.
- d. Data that is electronically transferred to the Customer is identified and defined as follows:
 - (1) "Working" - work in progress, not formally submitted or made accessible; provided for information or communication; subject to internal CM (version control).
 - (2) "Released" - CM controlled version released or made accessible after internal interview and approval; selectively provided to or accessed by the Government or receiving party.
 - (3) "Submitted" - CM controlled master version formally submitted or made accessible to the Government or receiving party.
 - (4) "Approved" - CM controlled master version approved by the Government or receiving party.
- e. Records are kept for each data transaction.

Data items submitted to the library require that the librarian acknowledge receipt, if required, and distribute software or project technical data to the designated reviewers in a timely manner. Distribution of technical data is available in hard copy or electronic format. The due date for submitting comments or recommendations against the item under review must be within 30 days after the library date of receipt. The librarian will generate a cover memo or attach a message to indicate the due date for submitting review comments. The librarian will act as the focal point for receiving and collating all review comments prior to submitting the review comment package to the originator.

5.3 INTERACTIVE ACCESS TO DIGITAL DATA

Access to digital data will be controlled and maintained by the SCM automated CM tool. User access and user level permissions, working file version naming, re-identification of marked up versions and changed files will be administered by the Software Administrator. CSCI development, acceptance status and tracking will also be managed by the SCM automated CM tool. Similarly, access to status accounting data will be controlled in CSAS using operating system and database security administered by Configuration Management as directed by the cognizant Project Manager.

5.4 STATUS REPORTING

Data requirements are incorporated into the Configuration Status Accounting System (CSAS). The database is used to identify all CDRL data, to prepare status reports, and to track approval history. The database contains each contractually required data item and information on data submission. Listed below are the main areas addressed in the status reports.

- a. Data deliveries completed in the previous period.
- b. Data scheduled for submission.
- c. Data due but not yet delivered.
- d. Status of delinquent data.

5.5 DATA SECURITY AND CLASSIFICATION MANAGEMENT

Data security is an integral part of data management. Security requirements are considered throughout all areas of data management control and will be implemented as directed by Project Management or as stipulated in the contract.

6.0 CONFIGURATION IDENTIFICATION

Configuration Identification is defined as the currently approved technical documentation for a configured item as set forth in specifications, drawings, computer programs and associated documents. This documentation is systematically numbered for purposes of identification, retrieval and control. The Configuration Identification for the MRDOC Program consists of the approved Science Requirements Document (SRD), System Specifications, Interface Control Documents (ICD), Interface Design Specifications, Procurement Specifications, Program Package Documents, Acceptance Requirements, Acceptance Test Procedures, Acceptance Data Package (ADP), Software Design Documents, Version Description Documents (VDD) and engineering drawings and associated lists as detailed in the product baseline.

Federal Data Corporation will develop and originate all technical documentation required to define the functional and physical characteristics of the item. All documentation originated and/or utilized by FDC to manufacture the item will be customer approved. Upon approval by the customer, documentation masters will be maintained within FDC, under formal control, and stored and distributed, in accordance with FDC's internal operation methodologies.

Configuration identification is applied to all MRDOC Program software, both code and corresponding documentation. The Configuration Item (CI) is comprised of the actual produced software along with its associated documentation (specifications, design documents and program/procedure listings). The NASA Project Manager and the Configuration Manager determine the identification scheme that will be employed.

Configuration identification of computer programs and documentation during the development effort consists of established baselines and releases, as described in this plan, that are time-phased to the development schedules.

The following configuration items will be baselined under the PCS project and placed under SCM for the purpose of identification, release control, change control and status accounting:

Software Configuration Management Plan (SCMP)

The SCMP establishes the plan for the Software Configuration Management activities that shall be performed during the CSCI development phases of the MRDOC Program. It is responsible for the control of changes, the identification of individual CSCIs, various versions of the software, the auditing of the software configuration to ensure that it has been properly developed, and the reporting of all changes applied to the configuration.

Science Requirements Document (SRD)

The SRD scientifically defines the project's overall purpose, and objectives, and specifies the science requirements necessary to obtain those objectives.

Project Development Plan (PDP)

The PDP provides the necessary capabilities required of the proposed hardware throughout all phases of the project to accomplish the science requirements as defined in the SRD.

Hardware Capabilities Document (HCD)

The HCD identifies the science requirements and specifies the engineering requirements and future hardware capabilities. Operational requirements will be defined in the HCD by defining all operations after the system is complete and ready for service.

Software Requirements Specification (SRS)

The SRS describes the functional requirements for the software application. It includes Data Flow Diagrams, Data Dictionary, Functional Partitioning and Processing Narratives.

Software Development Plan (SDP)

The SDP translates the requirements from the SRS into the architectural and procedural detail necessary for implementation. It describes the System and Major Software Functions. It includes Data-Flow overview, Structure Chart, Interfaces, Procedure Design, File Structures and Requirements Cross-Reference Matrix. Software will be developed along with the hardware breadboards, prototype systems and flight systems. Software will be configuration managed and controlled during development. The software will be checked out, audited and verified completely in the prototype system prior to installation in the flight unit.

Software Test Plan (STP)

The STP describes FDC' approach to testing the software. It is typically divided into two parts: Unit Test and Integration Test.

Software Test Case (STC)

The STC will include a brief description of the test and its purpose, the module under test, the stubs and/or drivers necessary for the test, and any test data sets necessary to perform the test.

Software Test Report (STR)

The STR summarizes the results of executing the test plan

Module Source Code Listing

The actual listing of program source code of each individual module of the software project.

Executable Program

The "final" product, which is the functioning program that will be used to support the project subsystems.

System Operational Manual

User's manual for the operation of the system during ground and flight operations.

6.1 DOCUMENTATION IDENTIFICATION

Document identification numbers shall consist of three distinct segment identifiers, followed by the document's revision letter. The first segment is a three-letter project acronym that identifies the document as being part of a certain project. The second segment consists of a category abbreviation that identifies the type of document. The third segment consists of a four digit number, which will range from 0001 through 9999 and will be issued sequentially. The CM

Specialist will be responsible for assigning numbers to new documents. All approved documents shall be uploaded and maintained in CSAS by the CM Specialist.

A common numbering scheme is established for all documents created for release under the MRDOC Program. The following represents the programs presently in progress under the MRDOC Program.

- Document numbers will be assigned in accordance with the following format, which shall consist of four parts in process of final build.
- 1. **Project Name/Abbreviation** - CIR, FIR, LMM, MDCA, SAR etc.
- 2. **Document Type** - An abbreviation which categorizes the nature of the document, e.g. analysis, test procedure, assembly procedure, etc. Refer to the following listing for further definition of the document types.
- 3. **Document Number** - (0001-9999) - Document numbers shall be issued sequentially, used only once, and to avoid confusion, shall not be reissued for use on other document types.
- 4. **Revision Letter** - A-Z per standards.

Examples of document numbers: FIR-VDD-0001A, LMM-ANA-0001B

In reference to the first example above:

- **FIR** Document written for the FIR project.
- **VDD** Document is a version description document.
- **0001** Number indicates it is the first document.
- **A** Document's revision letter is A.

The following list of acronyms shall be considered standard for each systems application, as required.

<u>Acronym</u>	<u>Acronym Name</u>
ANA	Analysis Report
ASY	Assembly Procedure
DOC	Document
DES	Design
ENI	Engineering Instructions
ICD	Interface Control Document
IDD	Interface Definition Document
LST	List
MOU	Memorandum Of Understanding
MUA	Material Usage Agreement
PLN	Plans
PRO	Procedure
REQ	Requirement Documents
RPT	Report
SCD	Specification Control Document
SCH	Schedule

SPC	Specification
TMR	Technical Memorandum
TPP	Test Plan/Procedure
TRT	Test Report
VDD	Version Description Document

The following represents the Project/System acronyms that shall be used for any new document generated for the applicable system listed. All numbers shall be issued sequentially for each system, regardless of the acronym used.

<u>System Acronym</u>	<u>System Name</u>
CIR	Combustion Integrated Rack (FCF)
COM	Common Hardware (FCF)
CVB	Constrained Vapor Bubble
FCF	Fluids and Combustion Facility (FCF)
FIR	Fluids Integration Rack (FCF)
LMM	Light Microscopy Module
LoC	Low Volume Fraction Entropically Driven Colloidal Assembly
MDCA	Multi-User Droplet Combustion Apparatus
MIP	Mission Integration & Planning
MRD	MRDOC (Program wide Exhibits I, II & III)
PCS	Physics of Colloids in Space II
PHa	Physics of Hard Spheres Experiments-2
SAR	Shared Accommodations Rack (FCF)
TSC	Telescience Support Center

6.1.1 Document Types

The abbreviated categories and definitions listed below should be used to determine the appropriate document type for MRDOC Program software documents.

ASY

Assembly procedures document how to assemble the hardware. Certain items such as sample cells have very stringent assembly controls.

DES

Software Design documents illustrate the concepts used, design and analysis of a particular software module. These documents contain Unified Modeling Language (UML) class, sequence, collaboration and state transition diagrams.

DOC

Documents such as project requirements and specification documents, etc.

PLN

Major project documents such as management plans, development plans, product assurance plans, safety data package plans

REQ

Software Requirements documents describe the functional requirements for the software application.

SPC

Software Specification documents relate information not normally contained within a design document. These documents may contain vendor supplied information and implementation details.

TPP

Test Plans and Test Procedures that document in a detailed step by step fashion, how tests will be performed and accomplished. Most TPP's will have a follow up test report (TRT). However, not all procedures will require a formal test report. In some cases, a report may only consist of a filled in procedure or check-off list.

TRT

Test reports will document the as-run results or any deviation from procedure for all tests performed. Every TRT should have an associated TPP. Not all procedures will require a formal test report. In some cases, a report may only consist of a filled in procedure or check-off list. These TRTs are extremely important for documenting tests that support the safety verification process.

VDD

The software Version Description Document (VDD) includes the current system build and tracking of the associated module versions. The build numbers are computer generated during the build process and are manually confirmed and entered onto the VDD. A project document identification number is assigned to the VDD to enable tracking successive changes to the build by SCO/SCN.

TMR

Technical notes will document everything else that does not logically fall under the other categories. Things such as compatibility data, advice or knowledge from lessons learned, design information, etc. will be documented here.

6.2 DOCUMENT RELEASE

Each document shall complete a cycle for review and signature approval prior to release (see Figure 8). If NASA approval is required, the document shall be processed in accordance with NASA's document review procedure. An electronic and hardcopy of the SCCB approved document shall be forwarded to the CM Library for retention and control and uploading into CSAS. A notification of document release will be distributed to affected functions within the organization.

6.3 SOFTWARE DOCUMENT REVISIONS

Software document revisions will be handled identically to that of changing software baselines. Software documentation including designs, specifications, requirements, etc. must be kept current and reflect the current state of the software as a whole. For this reason, software document revisions will coincide with that of software releases via Software Change Order/. Software Change Notice (SCO/SCN).

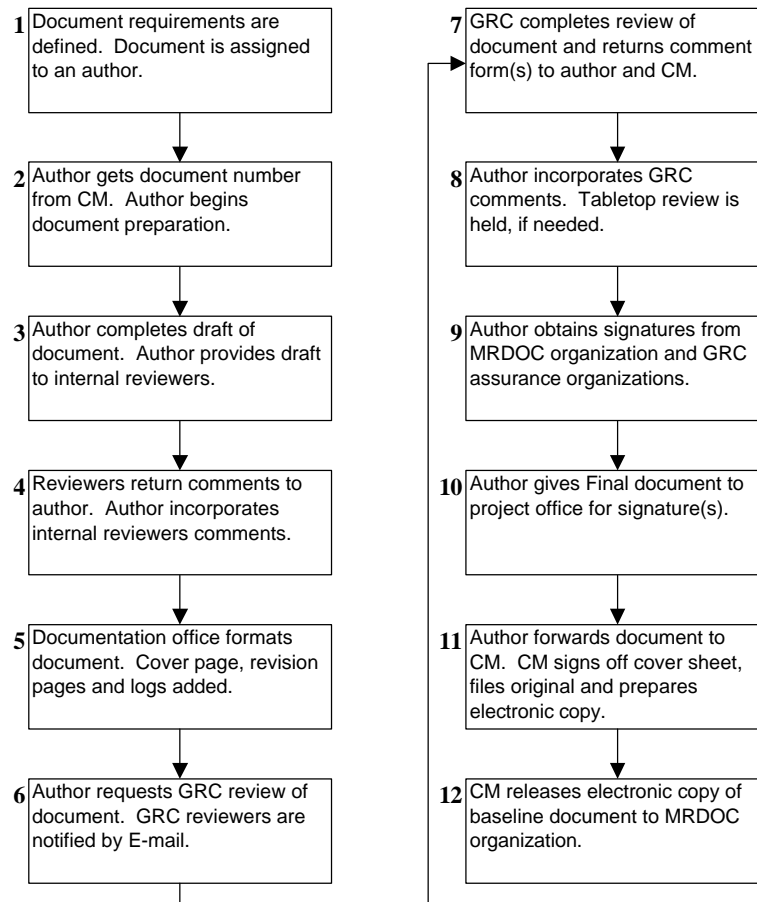


Figure 8 MRDOC Software Documentation Preparation & Review Flow

6.3.1 Software identification.

Software identification will consist of all technical data or information, including computer listings, regardless of media, that document the requirements, designs, or details of the computer software. The technical data explains the capabilities and limitations of the software, or provides operating instructions for using and supporting computer software during the software's operational life cycle.

6.4 SOFTWARE IDENTIFICATION

This section summarizes the labeling and version control methods to be used by the MRDOC Program for software identification and control. The technique FDC will employ and the level of control is dependent upon the maturity level of the software and phase of its development. Generally speaking, software module baseline versions and all subsystem software will be released to SCM and placed under formal control.

The MRDOC Program software system contains three subsystems. The flight software subsystem is the collection of software modules that execute on the flight unit. The laptop software subsystem is the collection of software modules that will execute on the EXPRESS laptop. The ground support software subsystem is the set of modules that execute on the ground support equipment in support of MRDOC Program. Software modules are the applications or processes that comprise the software subsystems. The version control systems and requirements defined in this document are for software modules and for software subsystems.

6.4.1 Software Modules

6.4.1.1 Preliminary Versions

Software developed in the initial phase of software development is called preliminary. This software is not generally of sufficient maturity to be of use to anyone except the developer of that software.

There is no requirement for formal version control for preliminary software. Therefore, FDC utilizes an automated software control tool as a repository to track the development history and to provide the capability to compare versions and revert to a previous version, as necessary.

6.4.1.2 Alpha Versions

A software module is considered to be at the alpha level of maturity when it is of sufficient maturity to be shared with other developers for their use. This software is not usually of sufficient maturity to be expected to operate flawlessly, and may include portions (functions) that are known by the developer to be incompletely implemented, or perhaps not implemented at all.

All software at the alpha level of maturity shall be controlled using an automated CM tool. The tool's labeling capability will be used to label the source code files using the module name followed by "version ALPHA M.m", where M shall be a decimal number associated with a

major release at the alpha level and m shall be a decimal number associated with a minor release. The first major release number for a module at the alpha level shall be 1; the first minor release number for each major release shall be 0. For each new major release or minor release, the release number shall be increased by 1. For example, the second minor release of the DataManager software for the third alpha version shall be labeled “DataManager version ALPHA 3.1”.

6.4.1.3 Beta Versions

A software module is considered to be at the beta level of maturity when it is of sufficient maturity that the developer believes that it is mostly complete. At this level of maturity, all problems identified with the software module are reported and tracked.

All software at the beta level of maturity shall be controlled using an automated CM tool. The tool’s labeling capability will be used to label the source code files using the module name followed by “version BETA M.m”, where M shall be a decimal number associated with a major release at the beta level and m shall be a decimal number associated with a minor release. The first major release number for a module at the beta level shall be 1; the first minor release number for each major release shall be 0. For each new major release or minor release, the release number shall be increased by 1. For example, the third minor release of the Correlator software for the second beta version shall be labeled “Correlator version BETA 2.2”.

6.4.1.4 Baseline Versions

A software module is considered to be at the baseline level of maturity when it is brought under formal configuration control. At this level of maturity, the developer considers the software to be complete. Changes for baseline software will only be accomplished through the formal CM control system utilizing a SCCB approved SCO/SCN.

All software at the baseline level of maturity shall be controlled using an automated CM tool. The tool’s labeling capability will be used to label the source code files using the module name followed by “version M.m”, where M shall be a decimal number associated with a major release at the baseline level and m shall be a decimal number associated with a minor release. The first major release number for a module at the baseline level shall be 1; the first minor release number for each major release shall be 0. For each new major release or minor release, the release number shall be increased by 1. For example, the first minor release of the Supervisor software for the second baseline version shall be labeled “Supervisor version 2.0”.

6.4.2 Software Subsystems

6.4.2.1 Beta Versions

A software subsystem is considered to be at the beta level of maturity when all of the software modules within the software subsystem have attained the baseline level of maturity. At this level of maturity, all problems identified with the software subsystem are reported and tracked. The only changes for subsystem software at the beta level will be in response to a formal SCCB approved SCOSCN.

All software subsystems at the beta level of maturity shall be controlled using an automated CM tool. The tool's labeling capability will be used to label the source code files using the subsystem name followed by "version BETA M.m", where M shall be a decimal number associated with a major release at the beta level and m shall be a decimal number associated with a minor release. The first major release number for a subsystem at the beta level shall be 1; the first minor release number for each major release shall be 0. For each new major release or minor release, the release number shall be increased by 1. For example, the third minor release of the flight subsystem software for the second beta version shall be labeled "Flight version BETA 2.2".

6.4.2.2 Release Versions

A software subsystem is considered to be at the release level of maturity when the software subsystem is considered to be suitable for use for the mission. At this level of maturity, all problems identified with the software subsystem are reported and tracked. The only changes for subsystem software at the release level will be in response to a formal SCCB approved SCO/SCN.

All software at the release level of maturity shall be controlled using an automated CM tool. The tool's labeling capability will be used to label the source code files using the subsystem name followed by "version M.m", where M shall be a decimal number associated with a major release at the release level and m shall be a decimal number associated with a minor release. The first major release number for a subsystem at the baseline level shall be 1; the first minor release number for each major release shall be 0. For each new major release or minor release, the release number shall be increased by 1. For example, the first minor release of the laptop software for the first baseline version shall be labeled "Laptop version 1.0".

6.4.3 Licensed Software

Licensed Software shall be given a company/project label with a unique identifier to indicate limited use. Periodic audits shall be performed by CM to determine adherence to the license limitations by the users.

6.4.4 Purchased Commercial Software

Purchased commercial software shall be relabeled with company/project identifiers and released for use and placed under formal CM control in the same manner as in-house software.

6.5 CHANGE CONTROL FORM IDENTIFICATION

Engineering Change Order/Engineering Change Proposal (ECO/ECP) and SCO/SCN forms are used to record and accomplish changes to baselined documents and software will be assigned a three segment control number to uniquely identify the change. This format will be similar to that for document identification specified in section 3.1.1. The first segment will be the three letter project acronym, "CIR, etc.", that identifies the change as being part of the CIR project. The second segment "ECO", "ECP", "DEV" or "WAV" identifies the change as being an Engineering Change Order/Engineering Change Proposal, Deviation or Waiver. The third

segment consists of a three digit number which will range from “0001 through 9999”. All change numbers will be assigned by the SCM Specialist.

Example: CIR-ECO-0001

6.6 CONFIGURATION BASELINES

This section defines the baselines and releases Federal Data Corporation will employ for the MRDOC Program. Baselines are established as a point of reference and control. The Project Managers, the Software Configuration Management Specialist and the Software organization will work as a team to establish and control:

- * A schedule of events and releases for each phase of the project. This will set activity guidelines to enable and ensure that the ultimate goals for each of the audits and reviews are met.
- * The formal release of all product baseline documentation that will assure the traceability and reliability of all engineering software changes.

Baselines establish and identify points of departure for the various stages (design, development, production and support) in the product life cycle. Baselines are established to define and control design, product, and engineering software and documentation changes. Baselines are time-phased to the development effort. Baseline implementation is determined by the NASA Project Manager. To provide progressive definition and design information of the CSCIs, the MRDOC Program will utilize five types of baselines: functional, allocated, developmental, product and production as shown in Figure 9.

6.6.1 Functional Baseline

The functional baseline is established by the acceptance, or customer approval of the project's requirements specification. This occurs at the completion of the Requirements Definition Review (RDR).

6.6.2 Allocated Baseline

The allocated baseline is established upon customer approval of the Software Requirements Specification (SRS). This corresponds to the completion of the Software Specification Review (SSR). The specification(s) and associated documentation contained in the functional baseline define the allocated configuration identification.

6.6.3 Developmental Baseline

The developmental baseline is established by the approval of technical documentation that defines the top-level design and detailed design including documentation of interfaces and databases for the computer software. This corresponds to the time frame spanning the Preliminary Design Review (PDR), the Critical Design Review (CDR) and up to the Functional Configuration Audit (FCA).

System/Software Development Life Cycle

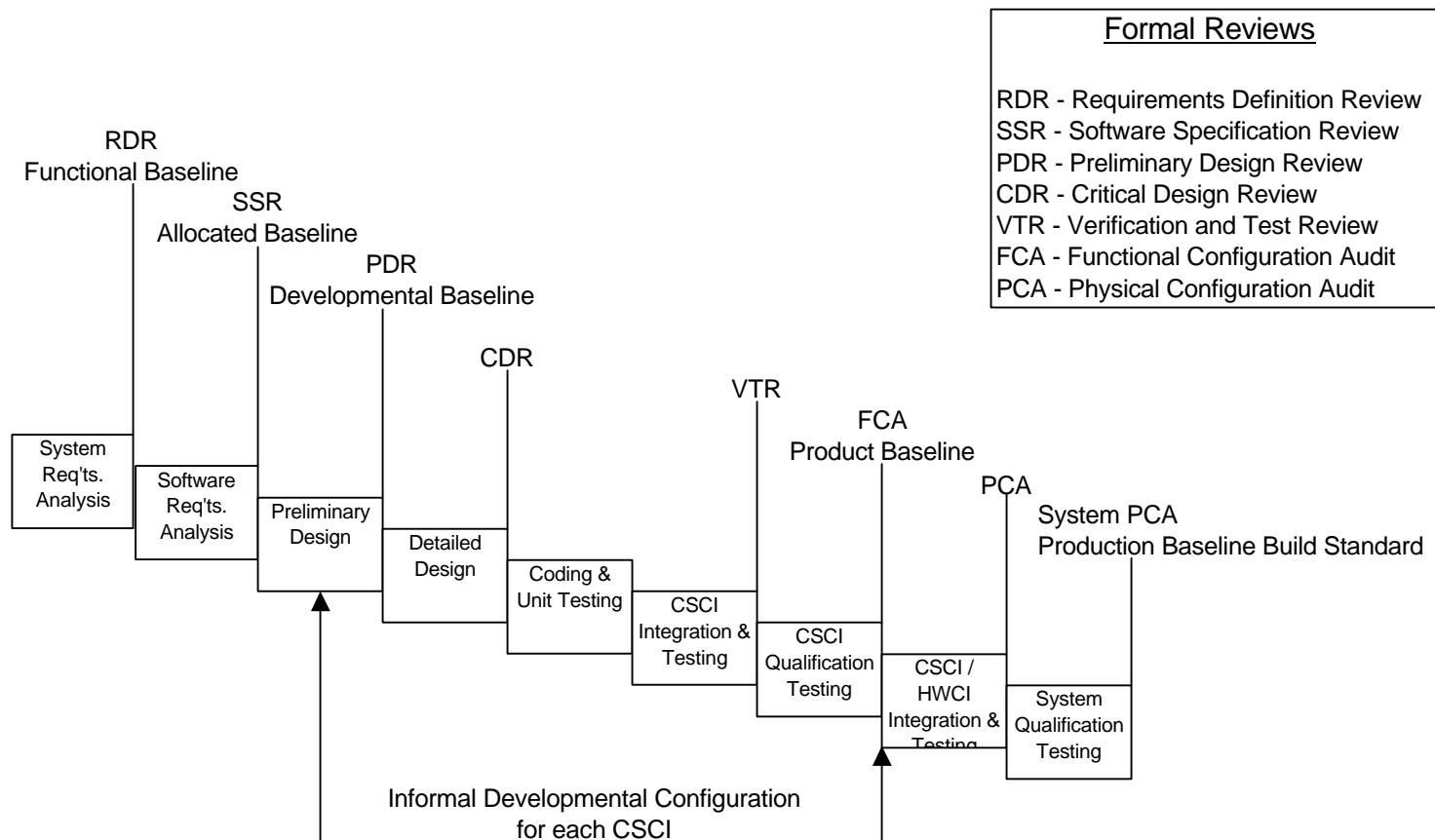


Figure 9

6.6.4 Product Baseline

The product baseline is founded upon the NASA approval of the product specification and is established just prior to the start of acceptance testing and the Functional Configuration Audit. Software versions will be correlated and documented through the use of a Version Description Document (VDD). After successful completion of the software acceptance testing, a VDD shall be developed and processed for approval, fully describing the software release information for each CSCI. This baseline is software equivalent to the production baseline for hardware, whereas this baseline defines the software build standards.

6.6.5 Production Baseline

The Production Baseline sets the build standard by defining manufacturing drawing issues, Contract End Item (CEI) Production Specifications, Software Product Specifications, Production Acceptance Test Specifications, etc., down to a level required to build and test production hardware/software. Once documentation is released and under formal CM control, all subsequent changes shall be identified and processed in accordance with the change process as outlined in this plan. This baseline shall be established prior to Flight Fabrication, Integration and Test.

6.6.6 Configuration Releases

Baseline releases shall be performed throughout the development life cycle. These releases fall into three categories, engineering, preliminary, and final. It is the responsibility of CM to concur with or establish the release, version, and update number identifiers. During the preliminary and engineering phases, on-going development versions, which are to be released (or promoted) to the next level of development, shall be maintained in the Software CM tool of the software engineer until the element reaches a maturity level to where its development is complete. Once the element's development is considered complete, the baseline version shall be formally released to the CM system (see paragraph 6.7.2).

6.6.7 Physical Item Hierarchy

Each baseline will establish a framework depicting the Physical Item Hierarchy of the software for each CEI deliverable under the MRDOC Program. This framework, which represents an indentured bill of materials, provides a means through which any end-item configuration can be efficiently managed. It provides the structure for defining, scheduling and controlling physical items, supporting documentation and engineering changes. A CEI compiled of software, is a hierarchy of functional statements descending down to the units of code, which perform those functions. A reference designation (code) will be assigned to each level of assembly and all revision controlled components. Establishing this relationship provides a status sort capability from the top function or program to all, or any selected sub-function or sub-module and their respective controlled units.

A physical item hierarchy defines the parent-to-child relationship between each document released for formal configuration control for hardware contained within the end-item or software modules and units-of-code contained in the project (see Figure 10).

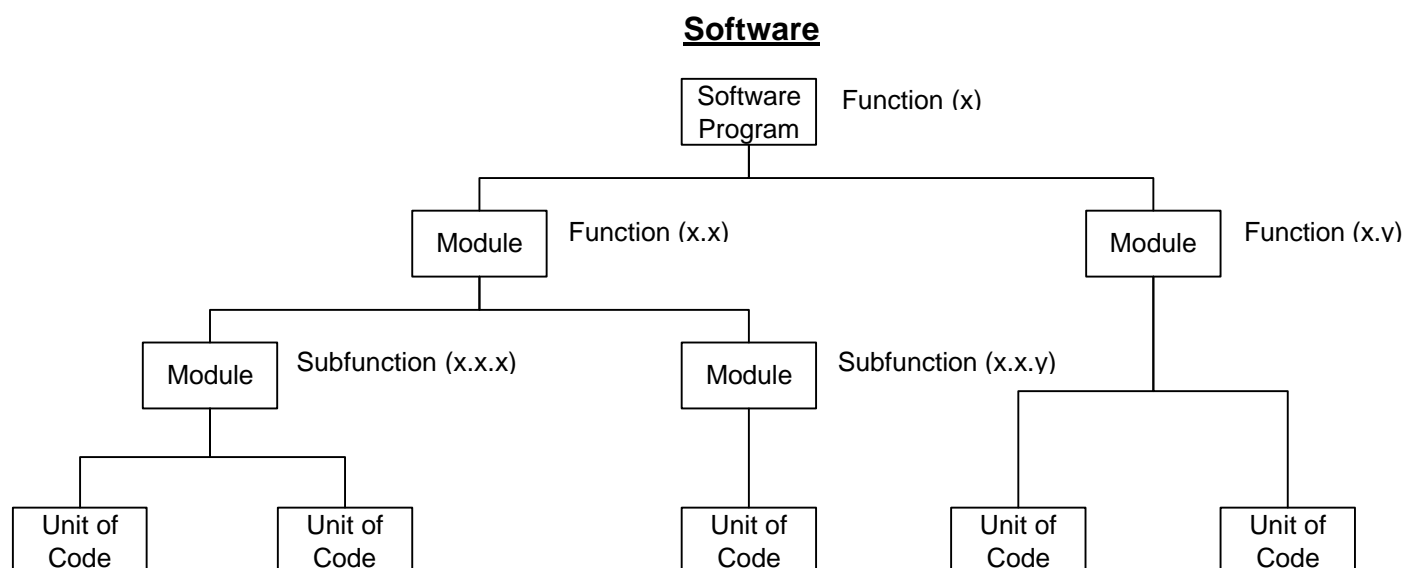


Figure 10 Hierarchical Structure for Software

6.7 SOFTWARE LIBRARIES

FDC will establish libraries throughout the software life cycle to maintain control and provide status accountability. These shall provide the means for identifying and labeling baselined entities and for capturing and tracking the status of changes to those entities. These libraries shall be comprised of the software files. The PCS project will administer four types of software libraries: development, integration, production and a repository.

6.7.1 Configuration Control of Libraries

The production software library and the software repository shall be released to and maintained by SCM. These controlled libraries shall be maintained in accordance with the change control processes defined in this plan. During the preliminary and engineering development phases, the development and integration software versions which are to be released (or promoted) shall be maintained by the cognizant software engineer and controlled in the software CM tool of the software engineering group until the element reaches a maturity level to where its development is

complete. Once the element's development is considered complete, the baseline version shall be released to the formal CM control system in the production software library.

6.7.2 Storage, Handling, and Release of Project Media

The software libraries shall furnish the necessary facilities for the controlled, secure, storage, handling and release of all project media. These libraries shall have limited access and shall be placed on the CM system. All MRDOC project media shall be impounded in these libraries. Personnel responsible for the handling and storage of software shall exercise precautionary methods to insure the prevention of any physical damage to, or degrading of the software elements. However, in the unforeseen event of damage or loss of the media, backups of the software will be archived separately to provide disaster protection. Two identical copies of the software release will be created. One will be used for installation; the other will be stored in a secure location as with the repository backups (see paragraph 6.7.6).

The software release is provided to the project's Lead Software Engineer. At this time, the release installation will be performed on the engineering hardware to test the general integrity of the release software. After evaluation, the release will then be installed upon the flight hardware under the supervision of the Software Quality Assurance (SQA) Engineer. Upon validation by SQA, a formal Software Release Notice will be signed off releasing the CSCI to the Production Software Library where it will be impounded and placed under formal control (see paragraph 6.7.5).

All released software will be accompanied by a read only CD-ROM disk made up of the following components:

- Installation Instructions
- Installation Utility
- Project Software Executables
- Operating System
- Project Software Source Code
- Vendor supplied libraries and source code (if applicable)
- Project Software Documents (VDD etc.)

The release notice will also report the following information:

- Release Identifier
- Date of release
- Version numbers, patch levels and dates of the compiler and operating system
- Version numbers and dates of the PCS source code modules

After baselining a CSCI, all future releases will be accompanied by a report detailing any SCO's/SCN's implemented in that particular release. This report will be distributed for management review and information.

6.7.3 Software Development Library

This type of library is also called a programmer's library, and it shall be used by the software system programmer(s) in developing their code. It is freely accessible to the programmer responsible for that unit at any time. Changes to software and configuration identification will be implemented by the cognizant programmer. Informal change and version control, software and historical data retention, as well as status accounting will be maintained in the software group's electronic CM database tool. Criterion for allowing promotions into the integration library includes the successful completion of unit testing and approval by the group's supervisor.

6.7.4 Software Integration Library

The integration library shall be used to capture and build the code that is designated for promotion to the test and evaluation process. This library contains the source code and executable load modules created as a result of a system build. The source code is placed in a special controlled library in preparation for a build. The code is then recompiled and link edited before it is placed in the integration library. Similar to the development library, configuration related activities will be provided and maintained by the cognizant software organization. The exception being a formal software inspection shall be performed prior to being promoted to and placed in the production library. Criteria for release to the production library includes:

- Submission of a software release notice by the SCCB.
- Completion of status accounting audits and resolutions of issues.
- Acknowledgment of regression and integration test completion by the test and evaluation group and approval by the systems Software Quality Assurance representative.

6.7.5 Production Software Library

The Production Software Library (PSL) shall contain the master copies of all the support software configuration items and computer program documentation including design and development notes, listings, software media, and Version Description Documents used on the project. Copies are made from the masters to be used on other systems. The production software library acts as a backup for the run-time configurations used by the systems.

The PSL is an integral part of the program engineering archival system and will be the repository for the following development materials:

- Computer program source code
- Computer program object code
- Computer program documentation including design and development notes, listings, software media, and Version Description Documents.

It will be the responsibility of the SCM Specialist to ensure that all design notes, source code, and software media that are released to the library have been approved and that access to all media and associated documentation is strictly controlled. Backups of the software will be archived separately to provide disaster protection.

For all software elements intended for use in the project's configuration, a change audit trail will be established and formal control will be imposed after an element has been baselined. Changes to these masters will be accomplished and documented via approved SCOs/SCN's.

The PSL will serve as a limited access or controlled storage area for those items that are to be verified, tested and submitted to the customer. All software to be loaded into MRDOC systems will be retrieved from the PSL. Formal configuration control begins when CSCIs are submitted to the PSL by Software Quality Assurance (SQA). A formal Software Release Notice (SRN) will be required for all releases of software. CM, upon receiving proper signatures on the SRN form, will store the SRN and associated software media/documentation in the project support library. All subsequent changes will be reflected in a new release of the software and SRN form. The SCM Specialist will then ensure that all documentation is complete and "lock" the media for that CSCI. SQA informal audits will be performed periodically to ensure that procedures are followed and no unauthorized changes are made to the CSCIs. Once a CSCI has been placed in the PSL, a copy can only be obtained upon the approval of SQA and the PCM. When the PCM receives an approved SCO/SCN for a CSCI, a duplicate copy of all material relating to that CSCI will be checked out to the requestor. The Master Copy never leaves the library and is retained for historical purposes. Once the change has been completed, the CSCI will be forwarded to SQA for validation and will re-enter configuration control in a manner identical to software entering configuration control for the first time. SQA will review the change to ensure the change is complete and correct. Upon validation of correct implementation the SCO/SCN will be documented to reflect its completion and forwarded to the SCMA.

6.7.6 Software Repository

This library shall be located on the CM system and shall be used to archive the various baselines and items released for general use. This library is also where the master copies plus authorized copies of the computer program configuration items in operational use are maintained. Copies of these masters can be distributed upon request and approval. The software repository will be backed up on a daily basis to tape and stored in a fireproof vault. Weekly backups will be saved for one month, at which point monthly backups will be retained permanently.

Baselines will be labeled using the automated CM tool's labeling scheme. Each baseline will be labeled with a unique identifier containing the version number in the form of "Release 1.0". A report will be generated containing the module name, release date and version number used in creating the release.

7.0 INTERFACE MANAGEMENT

FDC recognizes that effective interface management is essential to the MRDOC Program. The coordination of interface documentation is vital to the build, test, integration and operation of flight hardware and software in order to ensure compatibility and interoperability among various hardware and software components in a system. In addition to knowing the proper configuration identification of approved changes, comprehensive status accounting must be maintained for all in-process changes to achieve a cohesive management of the product baseline. All changes that affect interface documentation, must be formally processed on a Class I ECO, for review and approval by all impacted organizations. Interface Control Documents (ICDs) will be strictly administered by Configuration Management to ensure that sufficient levels of controls are applied. Controls will be implemented on the initial release of an ICD and all future changes processed for review. This pertains to both internally or externally generated changes.

Internally, FDC has instituted a problem reporting process, which will identify an area of concern, prior to the generation of change documentation. The problem report process involves a check of all outstanding changes to determine whether or not this problem has been previously or is currently being addressed elsewhere. The problem report review cycle serves as a gathering point of information so that improvements not considered mandatory, can be formally and effectively addressed the next time an item is to be changed.

FDC's MRDOC team will implement a practical interface management program that effectively controls the interfaces between the software and changing hardware configurations. SCM will assist in defining the interface and interface control requirements of the CSCIs and support equipment, as applicable. Functional and physical interface definitions will be 100% complete by Functional Configuration Audit/Physical Configuration Audit, and will be presented as such at these reviews.

CM supports the interfaces between the SCCB and the CCB by recording action items affecting each interface and following up on them to see that they are accomplished in a timely manner. CM maintains configuration control of the specification and the standards that control the interfaces between the software elements of the workstations. The workstation's configuration must include both hardware and support software for each installation. This includes accounting for leased and licensed software used on personal computers and in the workstation. CM maintains the operating system configuration used in support of the MRDOC Program as a means of enforcing control of the interfaces with the applications program. The definition of the interfaces will be defined in the individual interfacing specifications. There are four types of interfaces that will affect the MRDOC Program. These are organizational, phase, software and hardware.

7.1 ORGANIZATIONAL INTERFACE

This interface is comprised of the various organizational elements including the interfaces between the different organizations that are involved with the product. These organizations

include but are not limited to, vendor to buyer, co-developer to co-developer, and project to project. Vendor and major subcontractor interface management is further defined in section 11.

7.2 PHASE INTERFACE

The phase interface contains elements, which include the transition between the life cycle phases of the product that are included in this plan. They are coincident with the transaction in the control of the product between the different organizations. The interfaces are defined as the transactions of the documentation, code, data, tools, and records, which are managed by CM.

7.3 SOFTWARE INTERFACES

These elements are the agreements shared between the computer program product and the other software entities (for example: operating system, utilities, and communication system). These agreements involve the structure and meanings assigned to the data passing and the operational coordination of the data and the results. The software being used might already exist or may be developed by another project, if this is true; all the documentation belonging to these interfaces is to be controlled by CM.

7.4 HARDWARE INTERFACES

These elements are the interfaces that are the agreements that are shared between the computer program product and the characteristics of any hardware in the environment with which the program product interacts. These agreements involve capabilities provided by the hardware and operations defined by the computer programs.

8.0 CONFIGURATION CONTROL PROCESSES

FDC Software Configuration Management will control the configuration of all flight and GSE software for which FDC has design responsibility, and those end items supplied to FDC as Government Furnished Property (GFP) firmware/software. This control system provides a means of establishing formal configuration baselines and a process by which NASA and FDC can communicate, control and direct implementation of changes to the baselines.

This section describes the forms, methods of review, and the implementation procedures used by FDC to ensure full configuration control of changes against flight and GSE software, generated internally as well as externally, and changes against FDC unique support equipment and internal manufacturing processes. Configuration control for the MRDOC Product Baseline involves the identification and review of any changes that could affect the MRDOC Program, its interfaces and the test equipment used by NASA and FDC to ensure the operability of the end item. These changes will be fully reviewed and coordinated with NASA through the SCCB. Changes affecting documentation, processes and software will be documented through the use of internal ECO/ECPs and SCO/SCNs.

All FDC generated Class I ECO/SCOs, ECP/SCNs and Major RDWs will be coordinated with the PCS Configuration Control Board (CCB) for review of potential hardware interface effects prior to release and implementation. Class II changes and Minor RDWs will be used to document minor changes to deliverable software and for changes affecting only FDC unique internal documentation, processes, firmware, and software. CM shall process all software related changes, regardless of classification, for coordination through the SCCB for review and approval.

- Changes initiated by FDC subcontractors will be submitted through the respective contracting officer on FDC provided forms. These changes will be reviewed and dispositioned by the SCCB in the same manner as a FDC internally generated change.
- Changes requested by NASA are to be formally transmitted through the appropriate Contracts Representative to the Vice President Operations Office. These changes will be forwarded through Configuration Control to the appropriate CCB/SCCB representatives for review and evaluation.

8.1 SOFTWARE CONFIGURATION MANAGEMENT SPECIALIST

The Software Configuration Management Specialist (SCMS) represents the focal point for the review of engineering changes. The SCMS coordinates with the Lead Engineer to determine if the change is processed as Urgent, Fast Track, or via the conventional SCCB cycle. The SCMS shall verify that the change document includes all required information and that pertinent data is inputted into the CM system. A workflow process as determined by CM and Engineering shall prescribe the change review cycle used.

8.2 CLOSED-LOOPED CONFIGURATION CONTROL PROCESS

FDC's change control processes provide a closed-loop system for managing changes to MRDOC documentation, requirements, and configuration items. Managed changes include Problem Reports, ECO/ECPs, Deviations and Waivers. This closed-loop convention involves key personnel that ensure the appropriate processes are implemented and followed. This assures the traceability and control of a change from its initial identification of need through review, approval and implementation into the product and affected documentation. The SCMS, SCCB, CCB, and CM master document control function, provide the discipline necessary to certify that all change activity has been properly accomplished and accurately recorded.

8.3 PROBLEM REPORT

The FDC problem report system is a formal multi-purpose utility used to (1) document software failures and non-conformances, (2) request engineering assistance in resolution of manufacturing and procurement issues and (3) propose a Class II engineering change. A problem report will be used to identify, control and disposition issues applicable to product baseline software. The documented FDC internal form will provide the data necessary for system management and control of:

- Description of failure/non-conformance or problem.
- Cause of problem.
- Item(s) affected/Item traceability (effectivity).
- Segregation of unsuitable or non-conforming software.
- Disposition or resolution.
- Applicable resultant corrective action(s).
- Verification of resolution/corrective action implementation.
- Historical data to provide consistency, prevent recurrence, and track trends.
- Implementation, tracking, processing, and approval of urgent changes.
- Manufacturing and procurement issues requesting engineering assistance and/or resolution.

Problem Reports will be generated and processed in accordance with FDC procedure P 5003. Refer to paragraph 9.2.2 for those processes unique to development software problem reporting. All problem report activity will be progressively tracked in an electronic database from initiation through completion and closeout. To maintain awareness, automated reports will be distributed listing problem report status and highlighting open action items and the individuals responsible.

Change report disposition/resolution can effect the processing of a change request. In accordance with NASA Product Assurance Instruction PAI 360, if the Engineering Change is a Class II, the problem report may be used as the vehicle to propose/initiate the change (refer to section 8.5 and Figure 12 of this plan). This provides the means to process a pre-release for incorporating the change into the software. However, the problem report will remain open pending the submittal and subsequent approval of a RDW or formal ECO/ECP to permanently

change the baseline. Problem report and document change control numbers shall be cross-referenced between the related documents in order to augment the tracking of follow-up tasks.

Problem reports pertaining to failure/non-conformance issues will require a determination of the root cause, a proposed corrective action that will preclude the failure/non-conformance from recurring, and verification that all required actions were properly implemented or accomplished, and documented accordingly.

8.3.1 Customer Initiated Problem Reporting and Corrective Action

Customer initiated Problem Reporting and Corrective Action (PRACA) reports per NASA PAI 440 will be written to identify non-conforming or failed post delivered flight software. Issues that cannot be resolved onsite will be returned to FDC for analysis and any necessary remedial/corrective action. Upon receipt of the discrepant material and the PRACA, FDC will place the item(s) in bonded storage and generate a corresponding internal problem report in accordance with FDC procedure P5003. PRACA and problem report control numbers will be cross-referenced for tracking and accounting purposes. Upon completion of analysis and review, the problem report will be processed to resolve the problem. All subsequent activity will be documented on the FDC problem report. Upon completion, the problem report will be closed out and the PRACA will be updated accordingly and closed.

8.4 URGENT ECO/ECP/RDW PROCESSING

When unforeseen circumstances arise, a project's cost or schedule could be adversely impacted if FDC waits for a formal document change to be processed and approved. To preclude these cost and schedule setbacks, FDC management, with the concurrence of NASA, has imposed a system to effectively control the baseline and the hardware/software and thus permit the processing and implementation of unapproved urgent changes. This Urgent Processing procedure provides the disciplines necessary to insure the orderly processing, identification, implementation and tracking of urgent changes. The procedure will have safeguards in place to prevent NASA from incurring any risks attributable to the incorporation of unapproved urgent changes. FDC will implement this action to effect all types of changes (ECO/ECPs, Deviations or Waivers) pertaining to all types of items (hardware, software and all related drawings and documentation). When immediate action is deemed necessary, an Urgent change will be initiated, processed and implemented as depicted in Figure 11.

MRDOC

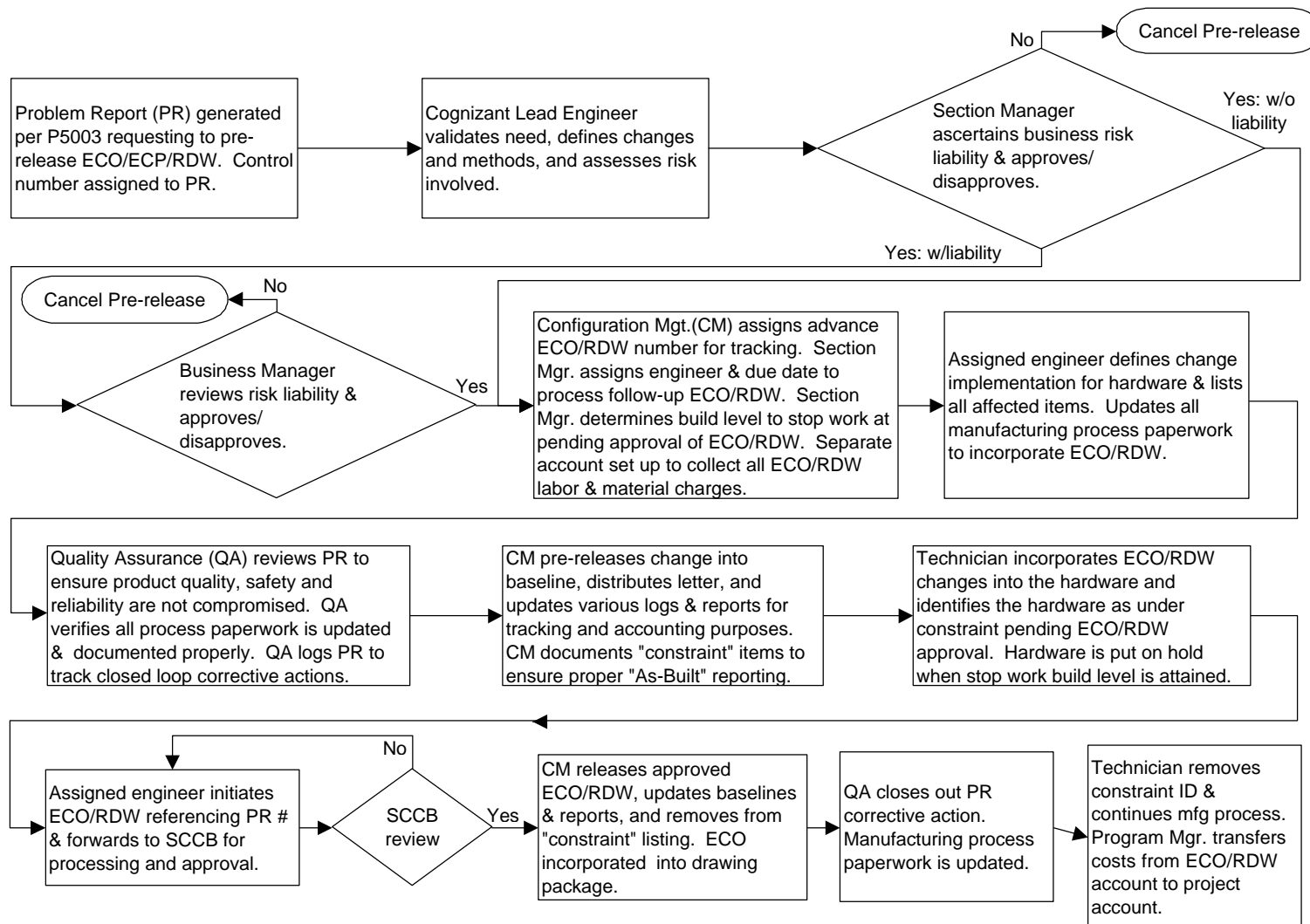


Figure 11 Urgent ECO/ECP/RDW Processing

8.5 FAST TRACK ECO/ECP/RDW PROCESSING

To improve the efficiency of the ECO/ECP review processing cycle, and minimize the impact on SCCB members, a system has been implemented for the effective review and approval of minor Class II changes. The Lead Engineer will review the change and determine if a formal SCCB is required. If the ECO/ECP is considered a minor documentation type change, the Lead Engineer will authorize approval. The ECO/ECP will also be distributed to SCCB members for their review. CM release will be held pending receipt of impact responses from SCCB members. If impacts are received, the change will be formally processed through the SCCB for assessment and approval signatures (see Figure 12).

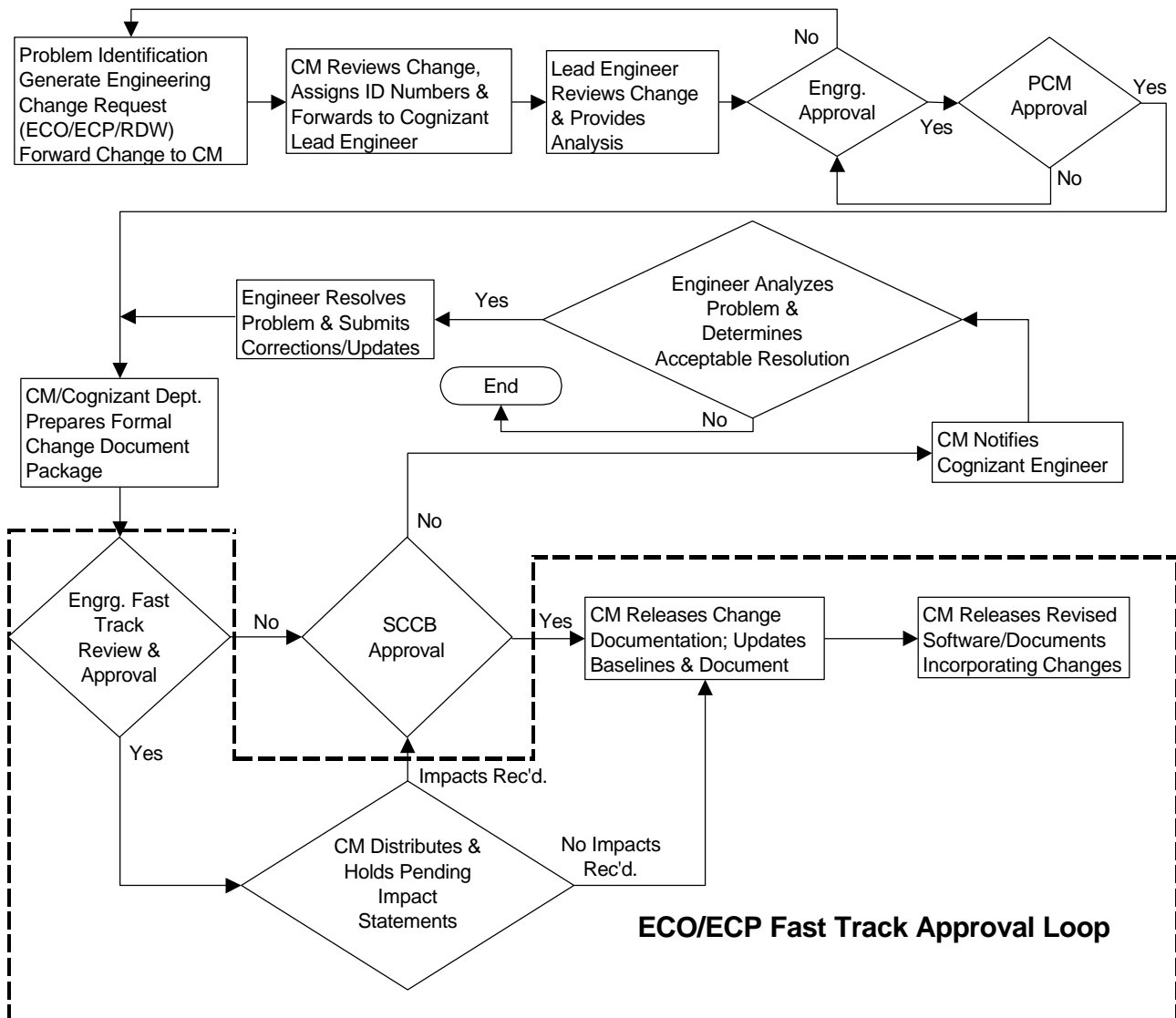


Figure 12 Change Control Process Flow Diagram (Class II ECO/ECPs, Minor RDWs)

8.6 CHANGE REQUEST INITIATION

Anyone at FDC may request a change. When problems with processes or enhancements to the product/testing are recognized by the area technician, Quality Assurance (QA), or production personnel, they will generate an Engineering Change Request (ECR), and the cognizant Lead engineer will review the ECR and add data or add further problem definition as appropriate. If the solution is obvious, an ECO/ECP or RDW may be written without going through the ECR process. All ECRs must receive the technical concurrence of the Lead Engineer before a formal change can be initiated. The Lead Engineer is responsible for reviewing the change for feasibility, technical content, and any potential for interface impact. After preliminary review, all changes are returned to CM for further processing. Problem statements and technical recommendations shall be stated clearly and concisely in a paragraph format. All affected documents shall be referenced. Results of tests, which prove the validity of the recommended solutions, shall be attached, as well as any cost and schedule impacts. Figures 11, 12 and 13 provide flow diagrams of the change control process tailored for use on the MRDOC Program.

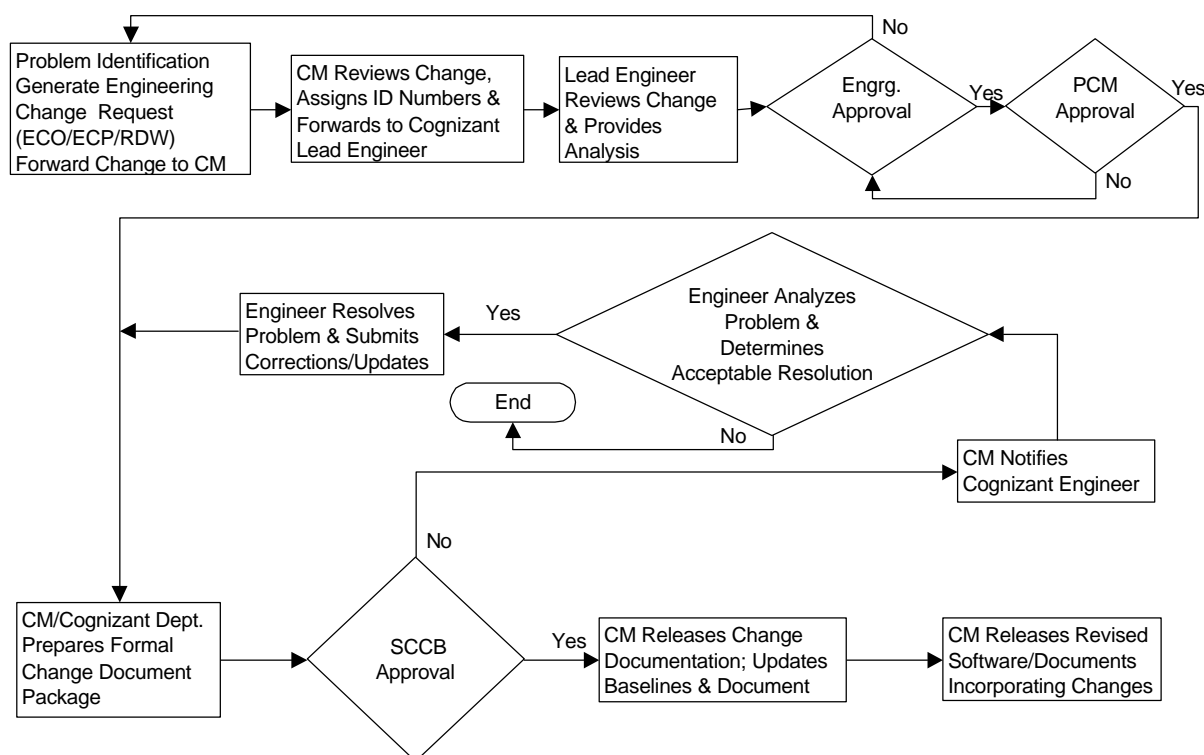


Figure 13 Change Control Process Flow Diagram (Class I ECO/ECPs, Major RDWs)

8.7 CHANGE CONTROL ORGANIZATION, AUTHORITIES AND ACTIVITIES

Software change control is governed by the SCCB. The Configuration Control Board (CCB) is responsible for hardware change control. The functional CM Group utilizes hardware and software specialists who ensure full accountability and control for proposed changes in their areas of expertise. Each specialist has the responsibility for ensuring that changes affecting both hardware and software are fully coordinated and when called upon by the PCM, will provide inputs to their respective review boards for proposed changes in their functional areas.

8.8 CHANGE CONTROL BOARDS

The Software Configuration Control Board (SCCB) will review and provide disposition on all software related baseline changes to controlled documentation. The Configuration Control Board (CCB) is responsible for hardware related changes. This plan will primarily discuss software and the SCCB. For further information in regard to hardware and the CCB, refer to the MRDOC Configuration Management Plan: MRD-PLN-0002.

Changes are submitted in the form of ECO/SCOs and ECP/SCNs. These changes will address the software as well as all baselined, controlled documentation. If necessary, a deviation will be submitted to intentionally depart temporarily from the mandatory requirements of a CSCI. A waiver will be processed if an item unintentionally does not conform to the configuration identification, providing the change does not impact the functional integrity of the end items. The SCCB must determine and recommend the proper classification of deviation or waiver (major, minor) and category.

Representatives assigned to the SCCB will be capable of viewing the overall effects of a change action. Included in this evaluation are determinations of impact on cost, schedule, baselines, product quality and reliability, safety, system interfaces and support activities. All change documentation must be reviewed for possible interface problems with other changes. Board members shall have full authority to act on behalf of their departments or functions and make recommendations or modifications to improve the clarity or technical approach to the change.

Board members do not vote. The SCCB chairperson has administrative control of the board. Based on the review and disposition inputs from the SCCB and in accordance with the requirements of the project, the chairperson will render a final disposition, either approving, rejecting, or directing the originator to amend the change package.

Individual function SCCB responsibilities are highlighted in the following paragraphs:

Software CM

The Software Configuration Management Specialist (SCMS) provides initial reviews of change documentation for format, compliance with governing directives and correctness of referenced

revision levels. The SCMA assigns control numbers to help track the changes through completion for entry into applicable manufacturing documentation and appropriate baseline.

Software Engineering

Included in this group are Software Requirements, Software Design/Development, Software Test and Software Systems Engineering. The Lead Software Engineer is responsible for all changes affecting application, operation and diagnostic software. Software Engineering will also provide protocol data for loading on-board and test equipment firmware.

Hardware Engineering

Hardware engineering disciplines will be represented by the Systems Engineer, Lead Electrical Engineer, Lead Mechanical Engineer and the Lead Optics Engineer. The hardware engineering group is responsible for the form, fit and function of all project hardware (including systems, subsystems and components) as well as the integration and coordination of firmware changes. They determine if a proposed change has any impact on hardware or software interfaces and if deemed necessary, the technical impact of incorporating the change into the hardware. Included in their assessment is the determination of the production effectivity or break-in-point. They are also responsible for the design and performance of FDC test equipment as well as project logistics and support impacts, such as spares, technical and training manuals, and user personnel.

Safety Engineering

Safety Engineering is responsible for assessing proposed changes to ensure that the inherent safety of the item, system or test equipment and that the system interfaces are not degraded. This assessment shall also ensure that there are no potential adverse effects on manufacturing, test, and operating and support personnel.

Quality Assurance (QA)

The QA engineer reviews all proposed changes for their impact on inspection and testing methods to ensure that complete inspectability and testability of the software, hardware and equipment is maintained at all times. QA ensures that changes are in accordance with FDC quality standards and MRDOC contractual requirements. The QA engineer will also evaluate impacts to overall operation, serviceability and reliability of the component assembly, end deliverable or test equipment for which the change is proposed. This analysis supports the logistics operation to assure that the level of maintenance and quantity of spares will adequately sustain support operations. The analysis will also identify any adverse impacts to the life cycle costs of the experiment.

Project Management

Project Management is responsible for the overall operations of the project. In this capacity the Project Manager will review all proposed changes to ensure material flow is maintained throughout the production process, including purchasing, production and inventory control, and processing. The Project Manager provides the SCCB with the status of units in process and determines how the proposed change affects the units that are finished, in process or not yet

started. In this manner, effectivity data is input as to when the proposed change break-in can take place, if additional personnel will be required to maintain schedule, and if additional tooling or facilities are required. He may request changes be initiated resulting from problems with vendors such as non-availability of parts, or vendor proposed changes that may have an effect on design requirements.

The Project Manager is also accountable for system engineering management of the project. In this function he is responsible for overall control of the system interfaces and design integrity. Any potential impacts attributable to proposed changes are immediately brought to the attention of the SCCB so that the problem can receive full review before any action is initiated. His review consists of the initial assessment of the proposed change's feasibility in relationship to project objectives, program impact, interface control and weight management. The advanced knowledge of potential change provided by the Project Manager to the SCCB members helps to eliminate any interface problems that may arise between test equipment and procedures and the latest revision of the affected item or end deliverable.

Ad Hoc Members

Ad Hoc members are called upon to provide increased depth of expertise where required. In this service, they are responsible for reporting on the effects a proposed change will have on their area of expertise. This includes representatives from other projects and programs within the company and suppliers or vendors who would advise on whether a change could be implemented in their product.

Business Operations

Business Operations will support SCCB functions by assessing and quantifying impacts for any changes affecting cost or schedule. If the SCCB review determines a change is out of scope of the present contractual obligation, and the impact affects cost and/or schedule, Business Operations shall review the change and apply appropriate impact towards FDC's management costs and build structure.

9.0 CONFIGURATION STATUS ACCOUNTING

The administrative tracking and reporting of all software items will be formally identified and controlled. This provides the data necessary to supply required analysis reports on all system interfaces and the identification and tracking of proposed as well as approved changes to the software baseline for the life of the project. It also provides technical inputs for definition of problem resolution by software problem reports for operational and test equipment software.

Configuration status accounting identifies precisely, at any point in time, the exact baseline change status for the system. SCM has implemented and will maintain a configuration status accounting system that will:

- Collect change status information from the appropriate organizational entity sufficient to reflect the status of all CSCI components under configuration control
- Maintain records on the status of all CSCI components and incorporation of approved changes.

Configuration status accounting reports will be used by management and the SCCB to determine CSCI status and to track change history. During the software development phase, status accounting for the evolving CSCIs will be maintained by the software organization utilizing that group's automated CM tool.

9.1 POLICY AND PROCEDURE

The Configuration Status Accounting (CSA) function for the CSCI is designed to:

- Provide and maintain a database for effecting traceability for each CI baseline in terms of its current configuration identification.
- Track changes to each baseline in terms of the task required to implement approved changes.
- Provide a management tool for tracking the accomplishment of changes through a series of reports.

9.2 STATUS ACCOUNTING CONTROL ITEMS

SCM will maintain accounting records to ensure full accountability of all configuration aspects of the MRDOC Program. SCM will utilize the MRDOC project Document Status Listing (DSL) to record configuration status accounting activities. The following status accounting control items will be provided and maintained by SCM:

- Part Number Record – The original entry records the part number and title for each CI under configuration control. Other information recorded includes the revision level or version and the release date. Subsequent updates require that the ECO/SCO, ECP/SCN numbers be entered along with the approval date, date of incorporation, new revision, effectivity and a brief description of the change.
- ECO/ECP Record – The record is used to track all associated ECO/ECP and their associated SCO, SCN, and NOR numbers, corresponding part numbers, date assigned, originator, approval or rejection date, a reference to related changes, and the status of implementation.

- Deviation/Waiver Record – The Deviation/Waiver record tracks the number assignment, originator of the request, date assigned, part number addressed, effectivity, description of change, and the disposition of the request.
- Software Problem Report Record – The problem report number is recorded and maintained in a log, which provides traceability for all proposed software changes.
- Version Description Document (VDD) – The software VDD includes the current system build and tracking of the associated module versions. The build numbers are computer generated during the build process and are manually confirmed and entered onto the VDD. A MRDOC project document identification number is assigned to the VDD to enable tracking successive changes to the build by SCO/SCN.

9.2.1 Document Status Listing (DSL)

A current listing of all documents related to software and CSCIs used by FDC to support the MRDOC Program forms the database for the computerized DSL. Reference designators (code numbers) will be assigned to illustrate the reporting structure of the CIs to their next higher assembly. The DSL is presented in a two-part format. The first part is a numerical sort by part number that includes the code assignment, document title, revision history (complete with revisions and corresponding ECO/SCO, ECP/SCN numbers, applicable RDWs, change documentation effectivities and a space for special comments or annotations. The second part of the DSL contains the identical information sorted by the reporting structure of the assigned codes.

9.2.2 Problem Reporting and Tracking

The practice of problem identification and reporting is necessary to permit an effective change control system and insure product quality. Accordingly, the tracking of problems is recognized as a fundamental component of status accounting. However, the level of formality applied should be commensurate with the maturity of the software. For example, during initial software development, the software is considered to be preliminary in nature. It is not of sufficient maturity to be of use to anyone except the developer of that software. Hence, formal problem reporting would not be necessary or cost effective. At this level of development, problem identification would be communicated using informal avenues such as e-mail, voice mail, hand written notes, etc. Once the software reaches a level of sufficient maturity and it can be shared by other developers, problem identification and reporting will be accomplished by one of two methods based on the nature of the problem.

- Minor issues or problems that are consequent to a lack of software maturity will be communicated informally applying the same means as preliminary software.
- Major issues or problems will be identified and tracked by the software organization using an automated tracking tool.

Configuration Management will not become involved in the problem reporting process until the software has been completed and the CSCI released to the Product Baseline. Once the item is placed under CM control, formal problem reporting and change processing will be invoked. The

processing and tracking of software problem reports will be accomplished as defined in section 8.3 of this plan.

10.0 CONFIGURATION MANAGEMENT DESIGN REVIEWS AND AUDITS

FDC shall provide a process for conducting effective configuration reviews and audits. The configuration reviews and audits scheduled for the MRDOC projects will establish the performance and functional requirements as defined in the configuration documentation. Their relationship in the software development life cycle is depicted in Figure 9. FDC will provide the necessary facilities, personnel, documentation and presentation packages, data sets, hardware demonstrations and any other functional requirement deemed necessary to support the required reviews and audits. The reviews will demonstrate that both the hardware and software meet baseline requirements, and that FDC's engineering release system is adequate to properly control the processing and formal release of configuration documents and changes. The following reviews and audits will be supported by Configuration Control.

10.1 REVIEWS

10.1.1 Software Specification Review (SSR)

The SSR will be performed to analyze the software requirements, validate the Software Requirements Specification (SRS) and ensure the SRS complies with the science, engineering, operational and functional requirements for the experiment. Successful completion of the SSR and approval of the SRS establishes the Allocated Baseline and normally signals the end of the validation phase and the beginning of the full-scale development phase.

10.1.2 Preliminary Design Review (PDR)

The PDR is a formal review of the system design and development approach, and will be conducted after the completion of the preliminary design synthesis, and before the detailed design process. The review will be conducted prior to the fabrication of an engineering model of the flight experiment. Successful completion of the PDR, as defined in the contract, will result in NASA's review board issuing the direction to commence software detailed design efforts and will establish the software development cycle and the informal Developmental Baseline.

10.1.3 Critical Design Review (CDR)

The CDR is a formal technical review of the project's design, and will be conducted after the design has reached the degree of completion needed to permit a comprehensive and detailed examination and data analysis. This review will be scheduled after the reviews of the engineering model system testing and prior to the coding and unit testing of the software and release of drawings and documents for fabrication of the end deliverable item. The objective is to have 90 percent of the flight documentation released at the time of the CDR. Successful completion of the CDR and closure of all resultant action items will provide the technical definition for the subsystem software and enable coding and unit testing followed by CSCI integration and testing.

10.1.4 Verification and Test Review (VTR)

The purpose of the VTR is to present the overall integration and test plan for the hardware and software and obtain approval from NASA to commence environmental and verification testing. The Software Verification and Validation Plan will be presented at this review. NASA's review board will assess the test plan and grant authorization to proceed with CSCI qualification testing efforts.

10.2 CONFIGURATION AUDITS

Configuration audits shall be conducted to verify that the as-built configuration of a Contract End Item (CEI) conforms to the as-designed requirements, as defined in the Product Baseline. Configuration audits will be performed at the conclusion of the definition phase and at the start of the build phase and includes a performance verification audit, the Functional Configuration Audit, and a design verification audit, the Physical Configuration Audit.

An audit team shall consist of a representative of each functional organization designated to participate in the audit planning, preparation and implementation. Audit plans and agendas are reviewed and agreed to prior to the audit. Audit of a complex product shall be accomplished in a series of incremental audits.

During the implementation of the audit, participants will record significant questions, discrepancies, non-conformances, and recommend courses of action. The audit team shall review the findings and determine appropriate actions. Affected parties will reach an agreement to the action items and establish a plan for closure. Minutes will provide a record of the audit findings, conclusions, recommendations, and action items. Follow-up shall occur until all required action items are complete. The necessary resources and material to perform an audit include the following items to the extent appropriate for the type and scope of the audit:

- Audit plan and agenda.
- Adequate facilities and unencumbered access.
- Assignment and availability of personnel.
- Applicable specifications, drawings, manuals, schedules, and design data, test results, inspection reports, process sheets, data sheets, and other documentation as deemed necessary.
- Tools and inspection equipment necessary for evaluation and verification.
- Access to the product(s) to be reviewed.

10.2.1 Functional Configuration Audit (FCA)

The functional configuration audit (FCA) is a means of formally validating that the allocated requirements, as defined in the CSCI development specifications have been satisfactorily completed and will perform as the design intended. FCAs shall be conducted for each software configuration item for which a separate development or requirements specification has been baselined. Successful completion of the FCA signifies the conclusion of software full-scale development and the initiation of the production phase wherein hardware/software integration and testing will be accomplished.

10.2.2 Physical Configuration Audit (PCA)

The Physical Configuration Audit (PCA) is a means of establishing that the CI/CSCI as-built, conforms to the Product Baseline and the engineering documentation for the item. This process is accomplished by comparing documentation with the hardware, software and firmware. PCAs verify that the released engineering drawings and product definition documentation adequately and accurately describe the as-designed configuration of the item, and the related manufacturing processes properly incorporate those requirements into the as-built configured hardware. The PCA will be conducted on the first articles of a CSCI. These items will be identified and selected jointly by NASA and the Contractor's PCA audit team. Destructive dismantling or disassembly is out of scope of the contract. Formal NASA approval and the satisfactory completion of a PCA result in establishment of the Product Baseline for the hardware CI and permit its progression to system level qualification testing.

10.3 CONTINUING PERFORMANCE AUDITS AND SURVEILLANCE

FDC shall perform informal ongoing reviews of production processes and procedures to ensure continued suitability and consistency of the product with its documentation. Operations of products or facilities shall be periodically reviewed to identify and monitor changes or degradation of performance, or to compare existing elements with new criteria or requirements. Associated documentation will be changed or updated when necessary to maintain consistency between the product and its definition.

10.3.1 Audits and Reviews of CM

To ensure that SCM efforts are adequate and completed as detailed in this document, audits and reviews of SCM processes and products are performed as described in the following paragraphs.

10.3.1.1 SCM Audits

To ensure that the SCM program complies with the requirements specified in this plan, an independent audit of SCM processes, procedures, and products is required. Normally, this type of audit is performed by a QA representative. Products generated or tracked by SCM are listed below.

- a. CSA reports
- b. Identified CSCIs
- c. Change requests
- d. Software version releases
- e. Libraries
- f. Documented SCM processes and procedures
- g. SCM review reports

The audit findings are documented in an audit report and provided to the SCM Manager. The audit report is used by the SCM Manager to correct deficiencies or identify changes in the SCM

requirements. Correcting deficiencies would include updating SCM processes and procedures, records, configuration documents, software, or tools. Identifying changes in the SCM requirements would result in adding, modifying, or deleting a requirement in this SCMP.

10.3.1.2 SCM Reviews

The CM Manager periodically performs internal reviews of SCM processes, procedures, and products. An SCM review serves as a method to determine how effectively and efficiently the SCM processes and procedures fulfill the SCM requirements as defined in this plan. SCM reviews also include verification of the products generated by SCM. Verification is the process of evaluating the products to ensure correctness and consistency with respect to the SCMP, tasks, processes, and procedures. The review findings are documented in a report that is used by the CM Manager to correct deficiencies or identify changes in SCM requirements.

It is the CM Manager's responsibility to perform or assign SCM personnel to perform the SCM reviews and to specify the SCM processes or procedures to be reviewed. The review report includes what actions were taken to resolve the deficiency or requirements change. The review report is filed and serves as a record to show that an internal SCM review was performed and corrective action was taken as required. Review reports may be audited.

11.0 SUBCONTRACTOR/VENDOR CONTROL

In the event software subcontractors or vendors are enlisted, FDC will invoke CM controls commensurate with the complexity of the deliverable furnished.

Subcontractors that design and/or produce equipment as a component of FDC systems, software programs, equipment or facilities, shall comply with the same configuration control requirements imposed on FDC by NASA, as applicable. FDC CM requirements to subcontractors/vendors will be issued through the following methods:

- Contracts
- Purchase agreements
- Formal correspondence and notification

11.1 SUBCONTRACTOR REVIEWS AND AUDITS

- Data reviews shall include assessment of supplier plans, procedures and configuration documentation.
- Configuration change management shall include review of proposed changes to buyer approved or imposed configuration documentation.
- Design reviews shall assess the seller's progress and provide a level of confidence that the product, when developed, will meet its specified attributes.
- Product test results shall be positive or negative indicators that required attributes that will or will not be satisfied.
- Configuration audits shall verify that the required attributes have been achieved and the design of the product has been accurately documented.
- CM surveillance reviews shall verify continuing application of supplier CM processes.

11.2 MAJOR SUBCONTRACTORS

If other major subcontractors are used, in addition to FDC's present MRDOC team members, the new subcontractors may be subjected to additional configuration management and change control requirements. Depending on the complexity of hardware/software being produced, an evaluation of the major subcontractor's Quality Assurance processes and CM controls may be conducted. If it is determined that additional controls are required, the subcontractor may be required to prepare and submit a Configuration Management Plan to FDC for review and approval. The outline and format of the plan is at the subcontractor's discretion, as long as it is in accordance with MRDOC contract requirements. The subcontractors would be required to define and describe their engineering drawing/software change control and release system and their plans for implementing a configuration management program in response to the work statement requirements in the areas of configuration identification, control, and accounting.

Any changes to the approved Configuration Management Plan are to be submitted to FDC for approval.

Software Changes shall be classified in accordance with the requirements of this plan.

- Class I changes shall be submitted to FDC for concurrence and processed through the SCCB for review and approval.
- Copies of Class II changes are submitted to FDC for information purposes.

All changes to items designated as flight significant (critical flight items) in the subcontractors work statement and procured by the subcontractor from a sub-tier supplier shall be classified as Class I changes and submitted to FDC.

FDC will periodically audit the suppliers CM system to ensure that the proposed methods are implemented for identifying and controlling each item of deliverable software produced.

Each subcontractor will establish a single point of contact, for communicating all CM issues between FDC Configuration Management and the subcontractor.

APPENDIX A ACRONYMS AND ABBREVIATIONS

A.1 Scope.

This appendix lists the acronyms and abbreviations used in this document.

A.2 List of acronyms and abbreviations.

<u>Acronym</u>	<u>Definition</u>
ABL	As-Built List
ADP	Acceptance Data Package
ANSI	American National Standards Institute
CDR	Critical Design Review
CDRL	Contract Data Requirements List
CEI	Configuration End Item
CEI	Contract End Item
CI	Configuration Item
CIR	Combustion Integrated Rack
CM	Configuration Management
CMM	Capability Maturity Model
COM	Computer Operation Manual
CN	Change Notice
CPM	Computer Programming Manuals
CSA	Configuration Status Accounting
CSAS	Configuration Status Accounting System
CSCI	Computer Software Configuration Items
CSEI	Computer Software Executable Images
DBDD	Database Design Descriptions
DM	Data Management
DOD	Department of Defense
DSL	Document Status Listing
DTP	Desktop Procedure
ECO	Engineering Change Order
ECP	Engineering Change Proposals
ECR	Engineering Change Request
EM	Engineering Master
FCA	Functional Configuration Audit
FCF	Fluids Combustion Facility
FDC	Federal Data Corporation
FIR	Fluids Integrated Rack
FPC	Functional and Physical Characteristics
FSM	Firmware Support Manual

Acronym

Definition

GFP	Government Furnished Property
GS	Ground Segment
GSE	Ground Support Equipment
HCD	Hardware Capabilities Document
HWCI	Hardware Configuration Item
ICD	Interface Control Document
IDD	Interface Design Descriptions
IRS	Interface Requirements Specifications
ISS	International Space Station
MIL	Military
MRDOC	Microgravity Research, Development And Operations Contract
MSD	Microgravity Science Division
NASA	National Aeronautics and Space Administration
OCD	Operational Concept Description
PAI	Product Assurance Instruction
PCA	Physical Configuration Audit
PCM	Program Configuration Manager
PDP	Project Development Plan
PDR	Preliminary Design Review
PIRN	Preliminary Interface Revision Notice
PM	Project Manager
PO	Purchase Order
PR	Purchase Requisition
PRACA	Problem Report and Corrective Action
PSL	Production Software Library
QA	Quality Assurance
RDR	Requirements Definition Review
RDW	Request for Deviation/Waiver.
SAR	Shared Accommodations Rack
SARGE	Standard Assurance Requirements And Guidelines For Experiments
SCCB	Software Configuration Control Board
SCM	Software Configuration Management
SCMA	Software Configuration Management Analyst
SCMP	Software Configuration Management Plan
SCN	Software Change Notice
SCO	Software Change Order
SCOM	Software Center Operator Manual
SDF	Software Development Files
SDL	Software Development Library
SDP	Software Development Plan
SIOM	Software Input/Output Manual
SIP	Software Installation Plan
SIR	Standard Interface Rack
SOW	Statement of Work

<u>Acronym</u>	<u>Definition</u>
SPS	Software Product Specification
SQA	Software Quality Assurance
SRD	Software Requirements Document
SRD	Science Requirements Document
SRED	Science Requirements Envelope Document
SRN	Software Release Notice
SRS	Software Requirements Specification
SSDD	System/Subsystem Design Description
SSR	Software Specification Review
SSS	System/Subsystem Specification
STC	Software Test Case
STD	Software Test Description
STD	Standard
STP	Software Test Plan
STR	Software Test Report
STrP	Software Transition Plan
SUM	System User's Manual
TPP	Test Plan/Procedure
TRR	Test Readiness Review
TRT	Test Report
TSC	Telescience Support Center.
UML	Unified Modeling Language
VDD	Version Description Document
VTR	Verification and Test Review
V&V	Verification And Validation

APPENDIX B DEFINITIONS

B.1 Scope.

This appendix defines special terminology used in this document.

B.2 List of definitions.

Term	Definition
Acceptance Baseline	The configuration baseline established as a result of the configuration inspections and acceptance reviews.
Acceptance Data Package (ADP)	The total data package for the Contract End Item (CEI) formally submitted to NASA upon final customer acceptance and delivery of hardware.
Allocated Baseline	Consists of hardware Type B development specifications and related documentation, and shall be established for each CI before the formal Critical Design Review (CDR).
Allocated Configuration Documentation	Documentation that describes the CIs functional, performance, interoperability and interface requirements that are allocated from a system or higher level Configuration Item.
Archive Library	A formal repository established for the control and archiving of all documentation and software created and released for use on the project.
As-Built Configuration	A numeric list of part numbers and revision status of all verified documentation, under configuration control, used in the build and test of a completed assembly.
As-Built List (ABL)	A listing of the configuration controlled part numbers, serial numbers and the revision levels, which define the configuration of the actual item built at the time of NASA acceptance.
As-Designed Configuration	A listing of engineering part numbers required for all CEIs and supporting documentation, based on the released engineering requirements, which reflect the configuration status applicable to the end-items at the time of delivery.
Audits (Software)	1.) A review of computer software performed by SCM to verify compliance with project requirements and baselines. Software audits may be informal, periodic audits, or scheduled formal audits such as a FCA or PCA. 2.) A regular investigation of the software installed on all computers in an organization to ensure that it is authorized or licensed. This minimizes software theft, and the risk of contracting viruses through uncontrolled copying. This practice also ensures technical support is available to all users.
Baseline	A configuration identification document or a set of documents, each of a specific revision/version and date, which defines the product configuration for a specific time during a CIs life cycle. The purpose is to provide a defined means to manage change. Baselines, plus approved changes constitute the current configuration identification.
Change	A modification to design, code or documentation, to correct an error or improve the system design and/or performance.
Change Control	A process that ensures all changes are properly proposed, communicated, evaluated, approved or disapproved, scheduled, tracked and implemented into documentation as well as the hardware/software.

Term	Definition
Classification of Change	An engineering change shall be classified as Class I or Class II in accordance with the requirements of MIL-STD-973. All classifications of software changes shall be reviewed and approved by the PCS SCCB.
Classification of Defects	Defects will be classified according to their seriousness and are normally grouped into the classes of critical, major or minor in accordance with the requirements of MIL-STD-973.
Computer Software Configuration Item (CSCI)	A configuration item that is computer software.
Configuration	The functional and physical characteristics of hardware, firmware, software, or a combination thereof as defined in the technical documentation and achieved in a product.
Configuration Baseline	Baselines represent configuration documentation formally designated at a specific point in time during a CIs life cycle. This documentation plus all approved change documentation constitutes the current approved configuration baseline.
Configuration Control	The systematic process that ensures all changes are properly identified, reviewed, approved, implemented, tested and documented, after the establishment of the configuration baseline for the CI/CSCI.
Configuration Documentation	Technical documentation that identifies and defines a Configuration Item's functional and physical characteristics.
Configuration Identification	The Identification includes the selection of CIs, determines the types of configuration documentation required for each CI, the issuance of numbers and identifiers to the CIs, including technical documentation that defines the CIs configuration, the release of CIs and their associated documentation and the establishment of Configuration Baselines.
Configuration Item (CI)	Any part of the development and/or deliverable system (whether hardware, software, firmware, drawings, inventories and/or documentation) which needs to be independently identified, stored, tested, reviewed, used, changed, delivered and/or maintained. CIs can differ widely in complexity and may contain other CIs in a hierarchy.
Configuration Management (CM)	A discipline applying technical and administrative direction and surveillance over the life cycle of items to: <ol style="list-style-type: none"> 1. Identify and document the functional and physical characteristics of configuration Items. 2. Control changes to configuration items and their related documentation. 3. Record and report information needed to manage configuration items, including the status of proposed changes, and implementation of approved changes. 4. Audit configuration items to verify conformance to specifications, drawings, interface control documents and other contract requirements.
Configuration Management Plan (CMP)	A document that defines what policies and procedures will be used to implement configuration management for a specific project, and which organizational structure is responsible for each phase of implementation. The Software Configuration Management Plan (SCMP) defines how software development and build activities will be managed.

Term	Definition
Configuration Status Accounting (CSA)	The recording and reporting of information required to manage configuration effectively. This includes a record of all approved documentation and identification numbers, along with all proposed, approved and incorporated change documentation.
Contract End Item (CEI)	The end deliverable line item, shipped via DD250, as defined by the contract.
Contractor Furnished Property	Items that the contractor is required to furnish in performance of a contractual agreement.
Critical Design Review (CDR)	A formal review of the detailed design of the CI/CSCI against the allocated requirements reviewed at the PDR/SDR. Data presented include all design documentation, specifications, test data, interface documentation, mockups and other related data. Completion of the CDR will result in authorization to complete the detail design and initiate the manufacture of hardware and the coding of software.
Data	Recorded information, regardless of medium or characteristics, of any nature, which can be assessed by a human or input into a computer, stored and processed there, or transmitted on some digital channel. Data by itself has no meaning, only when interpreted by some data processing system does it become meaningful information.
Database	A collection of related data stored in one or more computerized files in a manner that can be accessed by users or computer programs via a database management system.
Deficiencies	Deficiencies consist of two types: <ol style="list-style-type: none"> 1. Conditions or characteristics in any item which are not in accordance with the item's current approved configuration. 2. Inadequate or erroneous item configuration documentation that, has resulted, or may result, in units of the item that do not meet the requirements for that item.
Development	A formal process involving the analysis, design, coding and testing of software, where conceptual software requirements are progressively converted and matured into operational deliverables.
Deviation	A specific written authorization, granted prior to the procurement, manufacture, assembly or test of an item to intentionally depart from a particular requirement of an item's current approved configuration, for a specific period of time or number of units. This is an isolated or temporary condition that does not warrant a documentation revision.
Document Status Listing (DSL)	A baseline that represents a revision formal status of all approved documentation released at a specific point in time, for a project's CI/CSCIs.
Documentation	Any type of document, which is, or will be, subjected to master document control or formal change control. This includes drawings, policies, procedures, specifications and process plans, etc.
Effectivity	A point in time, event or product range (e.g. serial number or lot number) that specific products are affected by the implementation of changes or variances.
Element	A distinct part or subset of a computer program that performs a function or set of functions; a tool or set of tools used in the support of design, implementation or testing of a design.
Engineering Change	A change to the current approved configuration documentation at any point in the life cycle of the item.
Engineering Change	A FDC internal change form used for generating Class I and Class II non-software

Term	Definition
Order (ECO)	specific Engineering Changes for review and approval at the PCS CCB or SCCB.
Engineering Change Request (ECR)	A FDC internal change form used to request that an engineering change be processed against FDC documentation or for the generation of a formal Engineering Change Order.
Engineering Release	An action where configuration documentation or an item is officially made available for its intended use.
Firmware	The combination of a hardware device, computer instructions and/or computer data, that resides as read-only memory or programmable read-only memory on the hardware device. Firmware is easier to change than hardware, but more difficult than software on a disk.
Fit	The ability of an item to physically interface or interconnect with or become an integral part of another item.
Form	The shape, size, dimensions, mass, weight, and other visual parameters, which uniquely characterize an item. With software, form denotes the language and media.
Function	The action, or actions that an item is designed to perform.
Functional Baseline	The initial approved functional configuration identification (FCI), established by the authenticated system segment specification. Establishment is usually at the end of the conceptual phase.
Functional Characteristics	Quantitative performance parameters and design constraints, including operational and logistic parameters, and their respective tolerances. Functional characteristics include all performance parameters, such as range, speed, lethality, reliability, maintainability and safety.
Functional Configuration Audit (FCA)	The formal examination of functional characteristics of a configuration item, prior to acceptance, to verify that the item has achieved the requirements specified in its functional and allocated configuration documentation.
Functional Configuration Identification (FCI)	The currently approved or conditionally approved technical documentation for a configuration item, which defines: <ol style="list-style-type: none"> 1.All necessary and key functional characteristics. 2.Required tests to demonstrate specified functional characteristics. 3.Necessary interface characteristics with associated CIs. 4.Design constraints.
Government Furnished Property (GFP)	All property owned by the government, which is furnished to the contractor in performance of a contractual agreement.
Ground Support Equipment (GSE)	Mechanical and electrical systems developed to support the checkout, operation and turnover of flight hardware to the integrating facility.
Hardware	Products made of materials and their components, mechanical, electrical, electronic, hydraulic and pneumatic. Computer software and technical documentation are excluded.
Hierarchical Structure (Software)	The higher level and subordinate level (parent to child) relationships between various kinds of identified programmable elements. Structured in a top-down breakdown order from the element with the most dependent items/levels to the element with the least dependent items/levels.
Independent Verification and	The verification and validation of a software product by an organization that is both technically and managerially separate from the organization responsible for

Term	Definition
Validation	developing the product.
Inspection	Examination or measurement to verify an item or activity is in accordance with specified requirements.
Integrated Logistics Support (ILS)	A disciplined approach to the activities necessary to: <ol style="list-style-type: none"> 1.Cause support considerations to be integrated into system and equipment design. 2.Develop support requirements that are consistently related to design and each other. 3.Acquire the required support. 4.Provide the required support during the operational phase at a minimum cost.
Integration	Combining software or hardware components or both into an overall system that is compatible with each component and achieves the system objectives.
Integration Testing	Testing in which software and/or hardware components are combined and tested progressively until the entire system has been integrated meeting the system interface objectives.
Interchangeable Item	An item which possesses such functional and physical characteristics as to be equivalent in performance, reliability, and maintainability, to another item's similar or identical purposes and, is capable of being exchanged for the other item. This would be without selection for fit or performance, and without alteration of the items themselves, or the adjoining items, except for adjustments.
Interface	The functional and physical characteristics required to exist at a common boundary.
Interface Control	The process of identifying, documenting and controlling all functional and physical characteristics relevant to the interfacing of two or more items provided by one or more organizations.
Interface Control Documentation (ICD)	The interface control drawing or other technical documentation, which defines the functional and physical interfaces of related or co-functioning products.
Item	An all-inclusive term used to denote any product, including assembly, component, equipment, materials, modules, parts, subassemblies, systems or units.
Librarian	A position responsible for maintaining the project libraries/repositories. The librarian's responsibilities encompass managing the libraries, processing release notices, storage, maintenance and release of software, computer hardware and documentation, updating code, creating tape backups, etc.
Library	A limited access controlled repository used to store and archive software and related documentation applicable to the various phases in the development of a project. Back-up copies are also retained in a separate secure area for prevention of loss.
Life Cycle	The term covering all phases of acquisition, operation and logistic support of an item, beginning with concept definition and continuing through disposal of the item. The software life cycle typically involves requirements analysis, design, construction, testing (validation), installation, operation, maintenance and retirement.
Life Cycle Cost	The total cost to the Government of acquisition and ownership of a system over its life cycle.
Maintenance	The modification of a software product after delivery, to correct faults, to improve performance or other attributes, or to adapt the product to a changed environment.
Material	A term applied to systems, equipment, stores, supplies, spares, including related

Term	Definition
	documentation, manuals, computer hardware and software.
Media	Computer objects, on which data or computer software programs can be entered, stored and retrieved. These include hard disks, floppy disks, CD-ROMs, magnetic tapes, keypunch cards, read-only memory and programmable read-only memory.
Non-conformance	A condition of any article, material or service in which one or more characteristics do not conform to requirements.
Operational Test and Evaluation	Formal testing conducted prior to deployment to evaluate the operational effectiveness and suitability of the system with respect to its mission.
Operational Testing	Testing performed by the end user on software in its normal operating environment.
Part Number	A unique number applied to a part for configuration identification and accounting purposes.
Physical Characteristics	Quantitative and qualitative characteristics of material features, such as composition, dimensions, finishes, form, fit and their respective tolerances.
Physical Configuration Audit (PCA)	The formal examination of the CI/CSCI "as-built" configuration, to verify that a CI/CSCI conforms to the Product Baseline and its associated engineering documentation.
Preliminary Design Review (PDR)	A formal technical review of the preliminary design approach and allocations of systems requirements to the individual CIs/CSCIs. PDRs are conducted early in the design phase after the preliminary design has been completed and prior to the detailed design effort.
Product Baseline	The initially approved and/or conditionally approved documentation describing the form, fit and function requirements designated by the Acceptance Test Procedure for each CEI.
Production Baseline	In this baseline, manufacturing documentation requirements, CEI production specifications, software product specification and acceptance test procedures are defined down to a level needed to build and test production hardware and software.
Program	A single, complete and primarily self contained list of instructions, usually stored in a single file.
Quality	The totality of features and characteristics of a product that bear on its ability to satisfy stated or implied needs.
Quality Assurance (QA)	An internal organization that ensures a planned and systematic action plan of all steps necessary to provide adequate assurance that the CI/CSCI conforms to the established technical requirements.
Release	Documentation is considered released when it has completed the appropriate reviews and authorizations required for the specific phase for that document, and is considered suitable for use. After a formal release notification by CM, the document is subject to formal configuration control procedures.
Reliability	The probability that specific hardware or software items will operate without defect, within their designed limits, for a designated period of time.
Repair	A customer reviewed and approved procedure that reduces a nonconformance, but does not completely eliminate it. A repair is distinguished from rework in that the characteristics after the repair still do not conform to the established requirements.
Retrofit	The incorporation of new designed parts or software code, into items previously accepted and/or delivered to the customer. This necessity results from an approved Engineering Change to an item's current approved product configuration

Term	Definition
	documentation.
Reviews	The deliberate critical examination of the many facets of manufacturing processes and documentation. These are comprised of observations of the projects operation, the evaluation of procedures, audit results and the investigation of abnormal conditions discovered after the fact.
Rework	The procedure applied to a nonconformance that completely eliminates the discrepancy and results in characteristics that comply entirely with the established requirements.
Serial Numbers	A number that provides unique identification of individual components, assemblies and parts below the Contract End Item (CEI) level. Serial Numbers are considered for assignment primarily when there are traceability and interface requirements. Serial numbers are assigned consecutively and will not be repeated within a configured part number group.
Software	Computer programs and computer databases. Computer programs consist of a series of instructions, statements or rules, created in a format acceptable to computer equipment, designed to execute an operation or series of operations, or assert the rules in order to perform computational or control functions. Software can be split into two main types: system software and application software. Software includes both source code (written by humans) and executable machine code (produced by assemblers or compilers).
Software Change Order (SCO)	A FDC internal change form used for generating Class I and Class II software specific (computer software, computer programs and computer software documentation) Engineering Changes for review and approval at the PCS SCCB.
Software Change Request (SCR)	An internal FDC document used to request that an Engineering Change revision be generated for individual software documentation and other referenced documentation.
Software Configuration Control Board (SCCB)	A board comprised of technical and administrative personnel who recommend approval or disapproval or proposed engineering change documentation against CIs/CSCIs. This provides senior management visibility and review of a total change package for evaluation of the proposed change to determine if desirable, its potential impact on the system technical performance, reliability, safety, and program cost and/or schedule. The Configuration Control Board (CCB) has approval authority over hardware related changes, while the SCCB has authority for software related changes.
Software Configuration Management (SCM)	The discipline of configuration management as applied to systems or portions of systems that consist primarily of software. SCM invokes a set of engineering procedures for tracking and documenting software throughout its life cycle to ensure that all changes are recorded and that the current state of the software is known and reproducible. SCM consists of four components: Configuration Identification, Configuration Control (version control and change control), Configuration Audit and Configuration Status Accounting.
Software Development	A set of activities, that results in the development of software products. Software development may include new development, modification, reuse, reengineering, maintenance or any other activity that results in software.
Software Development Plan (SDP)	A management plan that defines the software development methodology. The SDP will govern the development process and ensure that the software media and imbedded programs are planned, analyzed, designed, coded, checked, integrated,

Term	Definition
	tested and delivered in accordance with contractual requirements.
Specification	A document that clearly and accurately describes essential technical requirements the for procurement of a product and provides a means to establish procedures necessary to determine that the product's performance meets it's requirements. A specification can be defined as a statement of specifics such as performance, characteristics, requirements and configuration for a given element of hardware or software.
Support Equipment	Hardware equipment and computer software required for development, testing, operating or maintaining a product item or facility in its intended environment.
System	A composite of equipment, software, skills and techniques capable of performing and/or supporting an operational role, which shall be sufficiently equipped for its operation and support, so that it can be considered a self-sufficient unit in its intended operational environment.
System Documentation	This documentation comprises the details of the design, the design philosophy, system capabilities, limitations and usage.
System Integration	The process of combining all hardware and software elements of a system, electronically, functionally and physically.
Technical Data	Recorded data, regardless of form, used to define, produce, test, evaluate, modify, deliver, support, maintain or operate a configuration item. Research and engineering records are included in this definition, but financial and administration records are not.
Technical Reviews	Reviews on the PCS project shall be a series of engineering activities, which assess the technical progress relative to the technical documentation or contractual requirements. Formal technical reviews, which will be conducted over the life of the software development and build process, shall be: <ul style="list-style-type: none"> • Software Specification Review (SSR) • Preliminary Design Review (PDR) • Critical Design Review (CDR) • Verification and Test Review (VTR)
Test	A determination and verification of an item's capability to meet specific requirements, as defined in the technical configuration documentation, by subjecting the item to a set of physical, chemical, environmental or operational conditions.
Traceability	Configuration traceability entails establishing and maintaining historical quality assurance data on components/assemblies procured and/or manufactured, as well as the location, where used parent application, of these items. Traceability will be applied through the controlled application and status tracking of: <ul style="list-style-type: none"> • Part Numbers • Serial Numbers • Lot Numbers • Log Numbers
Verification	A process used to substantiate and assure that an activity or condition has been performed, implemented or exists in conformance with the specified requirements. Verification methods include reviewing, inspecting, testing, checking, auditing and determining and documenting as applied to items, processes, services and documents. The verification process includes:

Term	Definition
	<ul style="list-style-type: none"> • Systems verifications to ensure developmental specifications meet the allocated system specification performance requirements. • Design verifications to ensure that as the design progresses, it continuously satisfies development specification requirements. • Informal testing of the hardware and software items. • Qualification testing to ensure that the end item satisfies the development specification requirements.
Verification and Test Review (VTR)	A formal review conducted prior to hardware/software integration for the purpose of reviewing the overall integration and test plan for the hardware and software and to obtain NASA approval to commence environmental and verification testing.
Version	A supplementary identifier used to distinguish a changed body or set of computer based data (software) from the previous configuration having the same primary identifier. Version identifiers are associated with CSCIs. A particular version is one of several sequentially created configurations of a product. Each version is usually identified by a number, commonly of the form X.Y where X is the major version number and Y is the release number. Typically an increment in X (with Y reset to zero) signifies a substantial increase in the function of the program or a partial or total re-implementation, whereas Y increases each time the program is changed in any way and re-released. Version numbers are necessary for effective configuration identification enabling the user to know if the program has changed and allowing the programmer to relate problem reports to the applicable versions. Problem reports and compatibility statements should always state the version(s) they apply to.
Version Description Document (VDD)	<p>A document file that provides precise data and information about the software media and accompanies a CI to support the computer software and convey the software's characteristics. The VDD furnishes data such as:</p> <ul style="list-style-type: none"> • Software identification, e.g. part number and program. • List of the materials provided. • List of changes incorporated. • Interface capabilities. • Applicable adaptation and installation instructions.
Waiver	A written authorization to accept an item, which during manufacture, or after inspection or test, is found to unintentionally depart from specific requirements, but is determined not to cause degradation if used, and is considered acceptable for use "as is", or after repair by an approved method.